AÑO III - NUM. 75

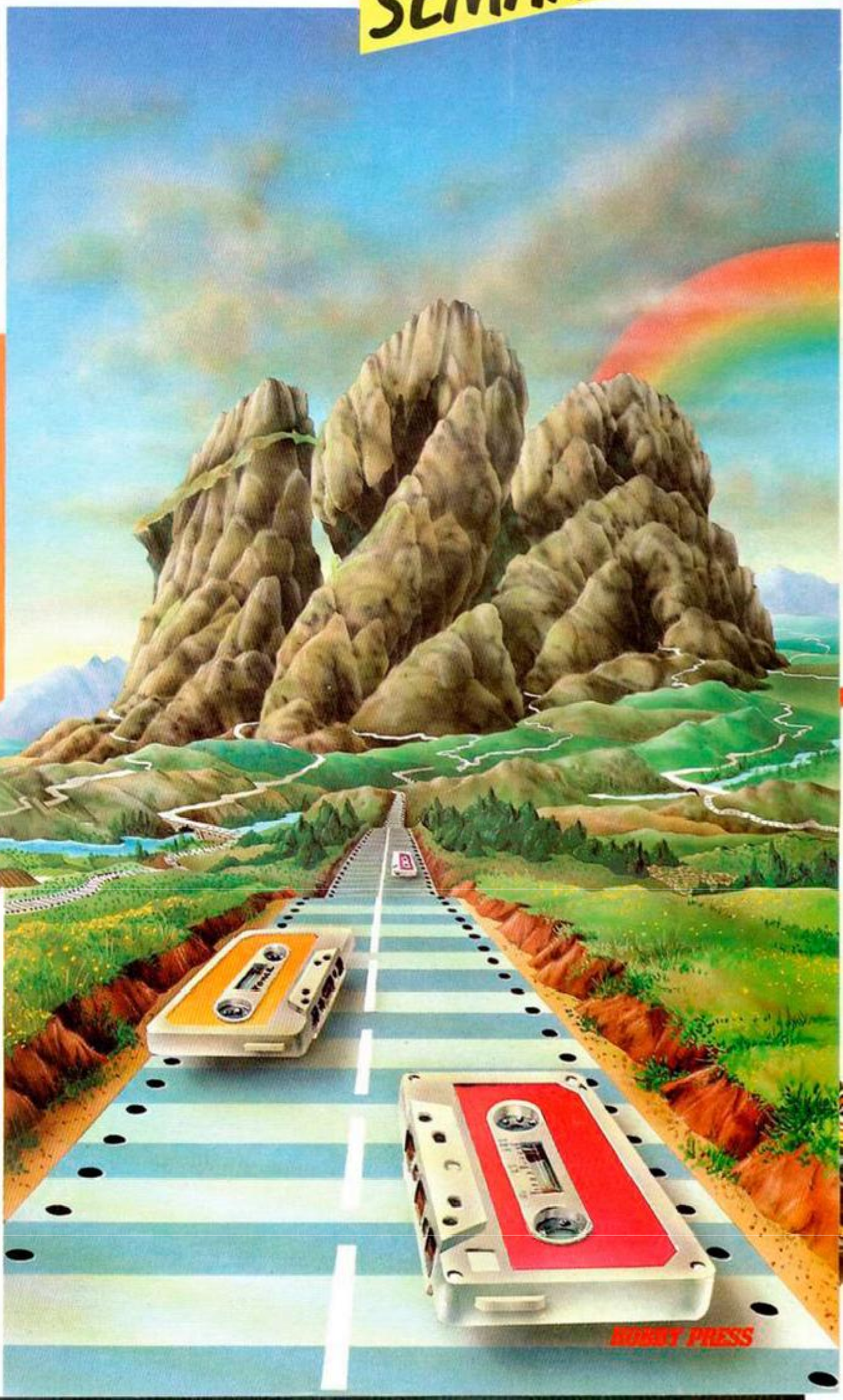
MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

SEMANAL

AÑO III - N.º 75

135 PTS.
Canarias 140 ptas.



MICROMANIA

ATENCIÓN USUARIOS ¡OJO CON LA VISTA!

INFORME

TODO EL SOFTWARE PARA EL 128K

NUEVO

BOMB JACK: Una misión para "Super-Ratón"

MICROMANIA

RESCATA A LA PRINCESA DE SIR FRED

TRUCOS

COMO ALTERAR EL ORDEN LÓGICO DE UN PROGRAMA


SPECTRUM
COMMODORE
AMSTRAD
BOMB JACK
ZAFI CHIP
¡INSTRUCCIONES EN CASTELLANO!
Después del tremendo éxito del programa en Japón, Bomb Jack ha pasado a ser el programa de arcade más famoso en USA y muy pronto lo será en toda EUROPA.
¡Diviértete!
Si están agotados
en tu tienda habitual
¡¡LLAMANOS!!
ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid.
Tel. 459 30 04. Tel. Bama. 209 33 65.
Telex: 22690 ZAFIR E
elite

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Redactora Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaría Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo

Corresponsal en Londres
Alan Heap

Fotografía
Carlos Candel

Portada
José María Ponce

Dibujos
Teo Mójica, F. L. Frontán, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Producción
Carlos Peropadre

Jefe de Publicidad
Concha Gutiérrez

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Paulino Blanco

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km 12,450 (MADRID)

Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

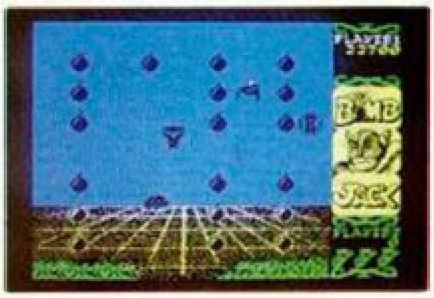
Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina)

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO III. N.º 75. 22 al 28 de abril de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.





Recorre el mundo con el superratón «Bomb Jack». (Pág. 12)

MICROHOBBY NUMEROS ATRASADOS

Queremos poner en conocimiento de nuestros lectores que para conseguir números atrasados de MICROHOBBY SEMANAL, no tienen más que escribirnos indicándonos en sus cartas el número deseado y la forma de pago elegida de entre las tres modalidades que explicamos a continuación.
Una vez tramitado esto, recibirá en su casa el número solicitado al precio de 95 ptas. + 6 de IVA hasta el n.º 36, a 125 ptas. + 8 de IVA hasta el n.º 60 y a 135 ptas. desde el n.º 60 en adelante.

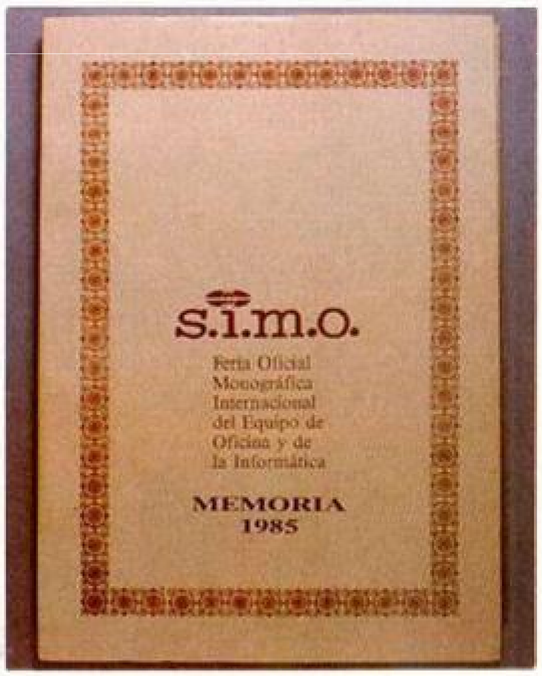


FORMAS DE PAGO

- Enviando talón bancario nominativo a Hobby Press, S. A. al apartado de Correos 54062 de Madrid.
- Mediante Giro Postal, indicando número y fecha del mismo.
- Con Tarjeta de Crédito (VISA o MASTER CHARGE), haciendo constar su número y fecha de caducidad.

Memoria SIMO 1985

PASADO, PRESENTE Y FUTURO DE LA FERIA



No cabe duda de la gran trascendencia que la feria anual del SIMO tiene actualmente y, hoy por hoy, a pesar de que se puede apreciar un ligero descenso del número de visitantes, así como del índice de operaciones y ventas realizadas, se puede afirmar que sigue siendo la feria más importante en nuestro país en cuanto a los temas relacionados con los equipos de oficina y de informática

Por esta razón, y aprovechando la circunstancia de que en esta edición el SIMO cumplía su 25 aniversario, sus organizadores han realizado recientemente una Memoria en la que se analizan detalladamente los resultados obtenidos, tanto en el presente año como a lo largo de su ya dilatada historia.

De ella se deduce que tanto los organizadores como los expositores en general, están de acuerdo en valorar positivamente la creciente adecuación de la misma a las necesidades de la oferta y la demanda en este campo.

En esta Memoria se dan a conocer, de una forma cuantitativa, los datos relativos a los tipos de materiales expuestos, así como el número de visitantes registrados en esta XXV edición.

De entre estas cifras, cabe destacar que en lo referente a los tipos de material, el que alcanzó un mayor porcentaje fue el dedicado al Tratamiento de Datos, que con su 33 por 100 sobre el total de lo expuesto, incluye una gran variedad de ordenadores, terminales y otros diferentes accesorios dedicados tanto al soporte de la información como al control del proceso de datos. Otros temas que también alcanzaron un índice elevado fueron los relativos a la Instalación y Comunicaciones (16 por 100) y Reproducción e impresión (12 por 100), donde se incluyen los diferentes modelos de impresoras, microfilms y archivadores.

Sin embargo, llama sorprendentemente la atención el bajísimo porcentaje de novedades presentadas, circunstancia esta que hace que la feria se convierta cada vez más en un escaparate donde los vendedores presentan sus productos, más que en un verdadero certamen donde los visitantes podrían comprobar el avance de la informática en nuestro país y en el extranjero. Este hecho se confirma con la circunstancia de que sobre un total de 174.000 personas que acudieron al SIMO, casi el 50 por 100 de las mismas procedían del mundo industrial y del comercio.



En cuanto al futuro de la feria, según una encuesta realizada entre los expositores, se presenta bastante incierto, y a pesar de que por regla general piensan que sigue teniendo porvenir, una gran mayoría de los mismos creen que sería necesaria una urgente reorganización.

En definitiva, de este informe se deduce la satisfacción por parte de los organizadores y expositores, pero quizá no vendría mal que este tipo de encuestas se realizaran entre los visitantes, pues seguramente de ella se llegarían a conclusiones notablemente diferentes.

LA INFORMATICA Y LA ARQUITECTURA

La informática va ahondando cada día más en el terreno de otras profesiones, con las cuales se establece una estrecha colaboración. Esto es lo que está ocurriendo en el caso de la Arquitectura, lo que se puede comprobar a través de dos acontecimientos que están llevándose a cabo en la actualidad: Los Encuentros del QL con la Arquitectura y el I Concurso de Carteles basado en el tema Informática para Arquitectos.

La primera de estas actividades se celebró recientemente en Expo-Sitap, Colegio Oficial de Arquitectos, y fue organizado por Investrónica, S.A. Durante estas jornadas se realizaron demostraciones prácticas sobre la utilidad y nuevas posibilidades que ofrece la información aplicada a la Arquitectura: redacción de proyectos, cálculo de estructuras...

La principal razón de haber elegido el QL como el protagonista de estos encuentros, radica en su sencillez de manejo en relación a las prestaciones que ofrece, por lo que prácticamente se hace innecesario cualquier conocimiento de programación.

Por su parte, SOFT y Hewlett Packard acaban de convocar un concurso de carteles para todos los arquitectos superiores españoles titulados entre el 1 de enero de 1979 y el 31 de diciembre de 1985 y los estudiantes matriculados en P.F.C. o 6.° Curso de Arquitectura.

El período de inscripción y presentación de trabajos finalizará el próximo día 25 de abril y el premio a dicho concurso será un ordenador personal Hewlett Packard 150-II y una extensa biblioteca de programas Soft para el mismo. Estos premios están valorados en 1.800.000 ptas.



Con este concurso se pretenden promocionar el acceso a los recursos informáticos por parte de los estudiantes de Arquitectura y últimas generaciones de arquitectos.

Mesa redonda organizada por Essilor

LOS EFECTOS DE LA PANTALLA SOBRE EL USUARIO

Recientemente se ha celebrado en Madrid, organizado por Essilor, una mesa redonda en la que se debatieron los problemas de la «fatiga visual de los trabajadores ante la pantalla». En la misma participaron diferentes especialistas en el campo de la oftalmología, la informática, la medicina de trabajo, así como el director general de Essilor en España.

Durante el transcurso del debate, los expertos discutieron, desde sus diferentes puntos de vista, los problemas que se producen por el hecho de realizar una jornada laboral frente a una pantalla de ordenador, del mismo modo que se constató el tema del creciente número de personas que invierten su tiempo libre en el hobby de los ordenadores personales.

De entre dichas conclusiones, pasamos a ofrecer un extracto de aquellas que por su temática, pueden resultar más interesantes a los usuarios de ordenadores personales:

— Los especialistas de la visión consideran que el nivel de desarrollo de un país está en relación directa con el número de gafas-usuarios. Así encontramos países como EE.UU. en el que usan gafas un 57 por 100 de la población, o un 55 en Gran Bretaña frente al escaso 37 por 100 de España.

— Está comprobado científicamente que las pantallas no emiten rayos ultravioletas ni infrarrojos. Las únicas radiaciones que emiten son de rayos X, las cuales no avanzan a más de 7 cm de la pantalla. Un ejemplo: una persona que permanezca 2.000 horas al año frente a una pantalla, recibe menos radiaciones que si pasara un fin de semana en la playa. Las pantallas no son nocivas en cuanto a radiaciones.

— Sí es cierto, sin embargo, que la larga exposición a las pantallas produce efectos comprobados: irritaciones oculares (54 por 100), aumento del pestañeo (50), dolores de cabeza (50), sensación de pinchazos oculares (46 por 100), y otros síntomas como deslumbramiento, quemazón ocular, ojos húmedos, etc... Por el contrario, otras molestias apuntadas como pérdida de apetito, caída de pelo, insomnio..., aunque indirectamente y psicológicamente relacionados con este tipo de trabajos, no son imputables a la pantalla.

— Los trabajadores ante pantallas, deben tomar por tanto, sus precauciones, tanto en lo referente a la pantalla en sí, como a la disposición del ordenador y sus periféricos. Estas se podrían resumir en los siguientes puntos:

— El teclado debe situarse de forma que quede perfectamente accesible, a un nivel confortable, con el fin de permitir a las dos manos operar de forma natural y en la prolongación del antebrazo.

— La posición de la pantalla debe ser a nivel de los ojos, con el fin de mejorar el confort de lectura. El brillo de los caracteres debe ser modulable para adaptarse a las necesidades de cada usuario.

— La ubicación de la pantalla debe estar alejada de factores de iluminación exteriores (ventanas, zonas aisladas, luces de neón...), pues éstas pueden ser factor de deslumbramiento.

— Nunca debe dirigirse una luz directamente hacia la pantalla.

— Nunca deben utilizarse lentes de sol.

— Unas persianas o cortinas pueden suavizar las luminiscencias exteriores, permitiendo un equilibrio general de los contrastes y evitando al ojo incesantes mecanismos de acomodación.

En resumen, estos debates sobre la ergonomía frente a la informática, han supuesto un avance más en el largo camino que deben recorrer las relaciones entre el elemento humano y el elemento tecnológico hasta llegar a una total compenetración.





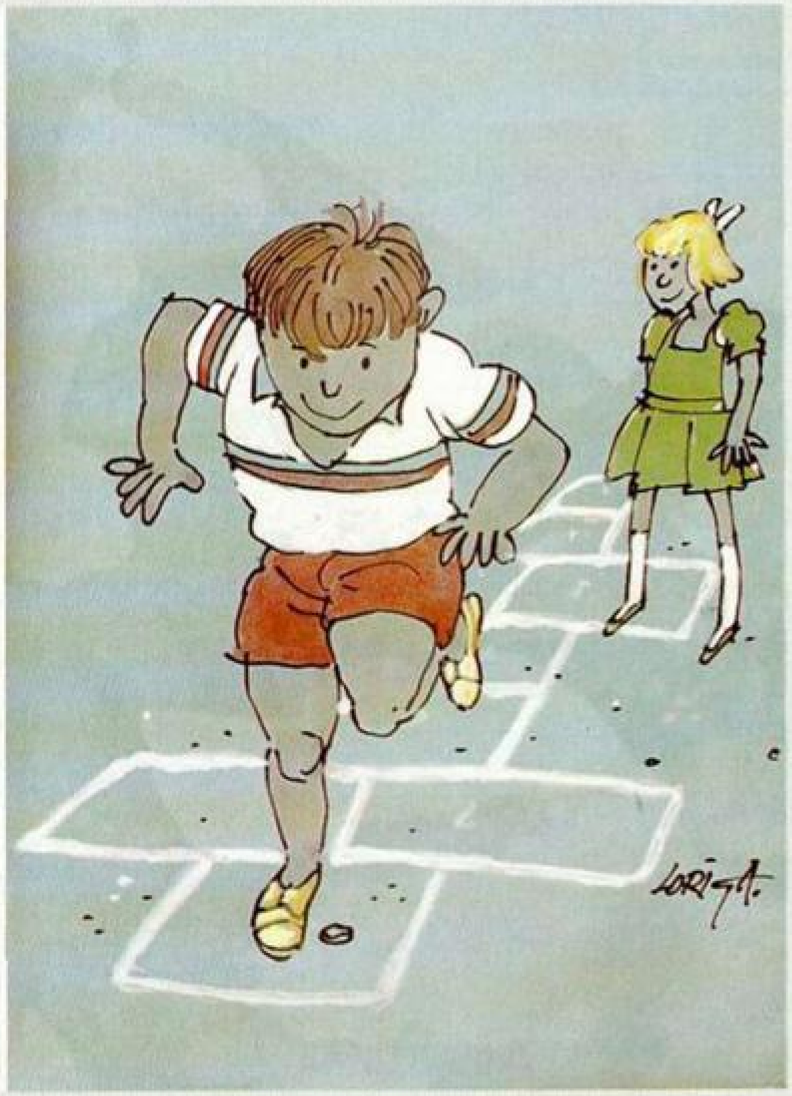
TRUCOS

PARA SALTAR... SENTENCIAS

Con este truco de José Jover tenemos la posibilidad de saltar alegremente a cualquier sentencia de cualquier línea, aunque la primera sea un REM. Esto se consigue operando directamente con POKES las variables del sistema que nutren la sentencia CONTINUE. Esta es la posición 23664, que indica el número de sentencia a saltar, y las posiciones 23662 y 23663.

Tecleando este pequeño programa será fácil de comprender introduciendo como número de sentencia el 2 y como número de línea el 2000.

```
100 INPUT "Numero de sentencia?
";a: POKE 23664,a
200 INPUT "Numero de linea ";b
210 IF b>9999 THEN GO TO 200
230 LET c=INT (b/256): LET d=b-
(c*256): POKE 23662,d: POKE 2366
3,c: CONTINUE
999 STOP
1000 PRINT "1": PAUSE 0: PRINT "
2": PAUSE 0: PRINT "3": PAUSE 0:
PRINT "4"
2000 REM xxx: PRINT "prueba"
```



ANTI - MERGE

Manuel Alfonso Martínez nos envía un truco para conseguir el ANTIMERGE total, de tal forma que si, por algún «curioso» se utiliza dicha sentencia, se dará en las narices al ver que el ordenador queda bloqueado, deleitándole con una simpática imagen. Para conseguirlo debemos añadir una línea al programa, no importa el número siempre que sea **al final** del mismo. Dicha línea contendrá:

9999 REM ANTIMERGE

A continuación, introduciremos en forma de comando directo: POKE (PEEK 23627 + 256*PEEK 23628)—13,0.

Con ello quitaremos la información de esa línea destinada a su longitud y cuando el ordenador intente introducirla en memoria, no sabrá responder.

Finalmente, salvaremos el programa con su correspondiente autoejecución.

EFECTOS DE BORDER

Este programa Código Máquina que nos ofrece Manuel Cordero, consigue unos efectos en el border de la pantalla y en el altavoz que, aunque son similares a otro truco publicado en el número 68 de MICROHOBBY, no se trata de la misma rutina. El programa usa la interrupción MODO 2, se ejecuta 50 veces por segundo y permite hacer otra tarea en el ordenador, vamos, que tenemos un programa dentro de otro.

Una vez que se tenga el programa C/M en memoria, hacer un NEW y activarlo mediante la instrucción RANDOMIZE USR 64000, y teclear el programa

```
10 CLEAR 63999
20 FOR a=64000 TO 64006
30 READ b: POKE a,b: NEXT a
40 DATA 62,250,237,71,237,94,2
01
```

```
50 RESTORE 80
60 FOR a=64255 TO 64279
70 READ b: POKE a,b: NEXT a
80 DATA 1,251,243,245,229,213,
197,6,255,197,120,211,250,16,252
,193,16,247,193,209,225,241,255,
237,77
90 BORDER 1: PAPER 1: INK 7: C
LS
100 PRINT #0;"PULSA UNA TECLA":
PAUSE 0
110 RANDOMIZE USR 64000
```

Programa Cargador:

En el listado ensamblador si se varía el número de la línea 18 en lugar de 150, cambian los bordes.

```
10              ORG   64000
20              LD    A,#FA
30              LD    I,A
40              IM    2
50              RET
60 ;
70 *D+
80 *L+
90              IM    1
100 *L-
110             RET
120             ORG   64255
130             DEFW  INIC
140 INIC        DI
150             PUSH  AF
160             PUSH  HL
170             PUSH  DE
180             PUSH  BC
190             LD    B,150
200 BUCL1       PUSH  BC
210 BUCL2       LD    A,B
220             OUT   (250),A
230             DJNZ  BUCL2
240             POP   BC
250             DJNZ  BUCL1
260             POP   BC
270             POP   DE
280             POP   HL
290             POP   AF
300             RST   #38
310             RETI
```

LAS CUEVAS DE SNAKE

Roberto QUIROS

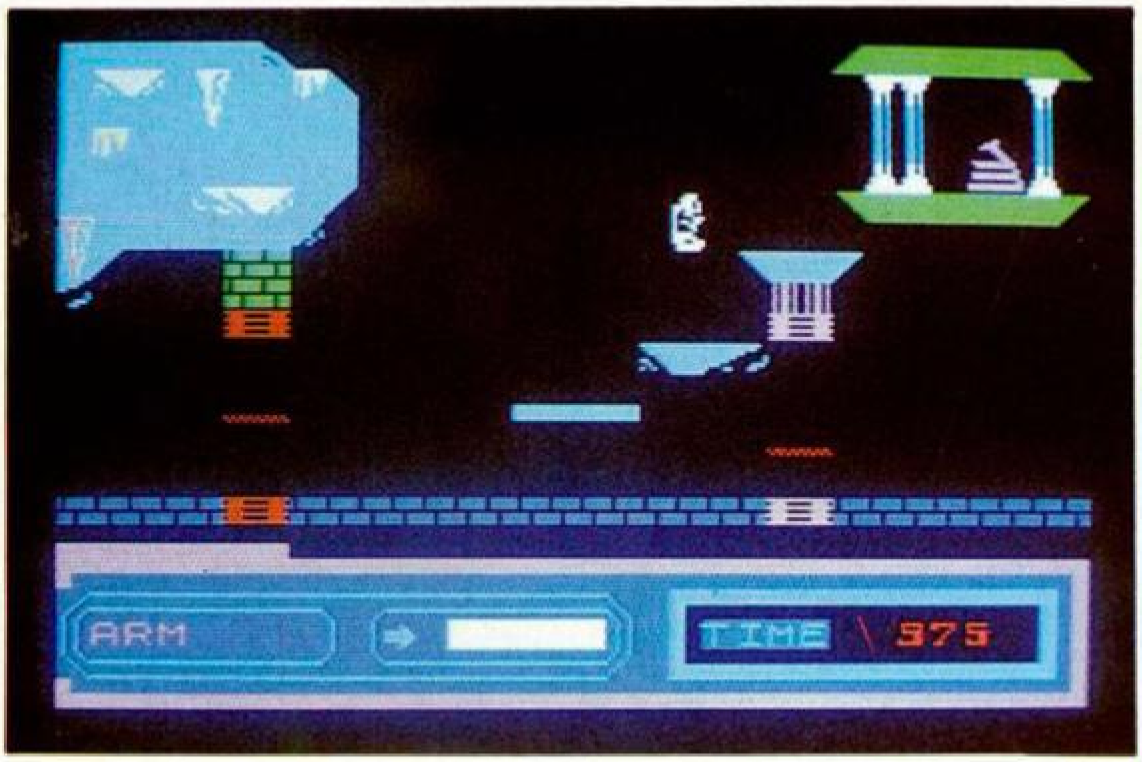
Spectrum 48 K

El misterio y la superstición rodea a las cuevas por todos conocidas como «Las cuevas de Snake», morada de la terrible serpiente guardiana del gran tesoro.

Muchos son los peligros que encierra este lugar, terribles sorpresas que nos hemos propuesto descubrir por nosotros mismos para conseguir el tesoro.

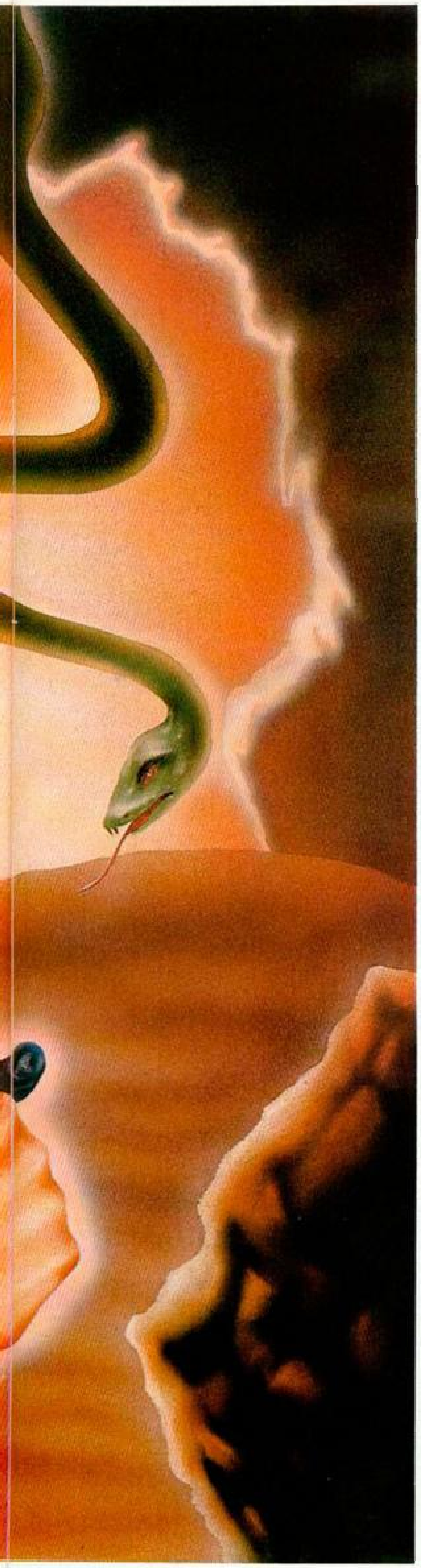
En primer lugar, debemos conservar nuestra única vida con la ayuda del «escudo mágico» que nos hará invulnerables y que encontraremos sólo cuando así lo dispongan los dioses. A lo largo de nuestro «paseo», daremos con «la espada de plata» con la que regresaremos a la primera pantalla para acabar con la «planta madre» consiguiendo, así, aniquilar todas las plantas maléficas que habitan en la cueva.

La «antorcha mágica», nos ayudará a superar otra difícil prueba hasta llegar a la pantalla del tesoro en la que aparecerá el arma capaz de destruir a Snake, el «garrote sónico». ¡Mucha suerte!




```
   1 REM UDG
   2 FOR N=USR "A" TO USR "D"+7: READ A: POKE N,A: NEXT N
   3 DATA 1,3,7,15,31,63,127,255,128,192,224,240,248,252,254,255,151,151,155,155,83,83,95,223,127,119,119,65,119,119,54,28
   5 LET G=1: LET HI=0
   7 POKE 23658,8
  10 REM INICIACION
  11 LET SC=0: LET ESCUDO=0: LET TIME=999: LET ANT=0: LET ESP=0: LET FM=10: LET GAR=0: LET BANT=0: LET PORC=0: LET HOJA=1
  12 LET B$=" PORCENTAJE  MALO ": LET AV=0: LET PAST=0: LET PANT=1: LET Y=14: LET X=10
  20 BORDER 0: PAPER 0: INK 0: BRIGHT 0: CLS
  50 REM PORTADA
  52 POKE 23607,60
  60 GO SUB 9800
  65 PRINT AT 4,5; INK 7; BRIGHT 1;"D LAS CUEVAS DE SNAKE D"
  70 PRINT INK 3;AT 7,5;" S  IZQUIERDA";AT 9,5;" A  SALTO IZQUIERDA";AT 11,5;" W  SALTO DERECHA";AT 13,5;" S  DERECHA";AT 15,5;" O  FUERZAS MAGICAS"
  75 PRINT INK 1;#0;TAB 20;"ROG ©1986"
  77 INK 5: PLOT 60,33: DRAW 134,0: DRAW 0,-11: DRAW -134,0: DRAW 0,11: PLOT 65,34: DRAW 130,0: DRAW 0,-12: PLOT 60,21: DRAW 10,0: INK 0
  79 IF G=1 THEN GO SUB 9000
  80 PRINT INK 4;AT 18,8;"PULSA UNA  TECLA"
  81 IF INKEY$="" THEN GO TO 80
  85 GO TO 95
  95 PRINT AT 18,8; INK 2; PAPER 5;"!!!! SUERTE !!!!"
  96 FOR N=1 TO 7: BEEP .2,30: BEEP .1,30+N: NEXT N: BEEP .5,37
  99 REM
 100 REM MARCADOR
 105 INK 0: PAPER 0: BRIGHT 0: CLS
 107 PRINT PAPER 1;AT 18,0;"                                "
 110 PRINT INK 3; BRIGHT 1; PAPER 1;AT 19,1;"ARM";AT 21,0;"";AT 17,0;"";AT 18,31;"";AT 19,31;"";AT 20,31;"";AT 16,0; PAPER 0;""
 120 PRINT INK 5; BRIGHT 1;AT 18,19;"";AT 19,19;"";AT 19,30;"";AT 20,19;""
 140 PRINT INK 5; PAPER 1; BRIGHT 1;AT 19,20;"TIME"; PAPER 0;" "
 150 PRINT INK 7;AT 19,12;""
 170 INK 4: BRIGHT 1: PLOT 8,8: DRAW 128,0: DRAW 6,6: DRAW 0,10: DRAW -6,6: DRAW -128,0: DRAW -6,-6: DRAW 0,-10: DRAW 6,-6
 180 PLOT 8,12: DRAW 56,0: DRAW 3,3: DRAW 0,8: DRAW -3,3: DRAW -56,0: DRAW -3,-3: DRAW 0,-8: DRAW 3,-3
 190 PLOT 80,12: DRAW 56,0: DRAW 3,3: DRAW 0,8: DRAW -3,3: DRAW -56,0: DRAW -3,-3: DRAW 0,-8: DRAW 3,-3
```








```
 200 POKE 23607,200
 220 PRINT INK 5; PAPER 1;AT 19,
10;"L"
 250 BRIGHT 0
 999 PRINT INK 7;AT 14,10;"I";AT
 13,10;"F"
1000 REM FANTALLA 1
1005 PRINT AT 15,0; INK 4; BRIGH
T 1;"::::::::::::::::::::::::::::
:::::"
1010 PRINT AT 0,6; INK 5;"██████
█████████████", INK 2;"::::"
1015 PRINT INK 5;AT 1,8;"███████
██████████";AT 2,15;"█████████
█ ███████";AT 3,28;"█████";AT 4,
29;"████";AT 5,29;"████";AT 6,29;"
███";AT 7,30;"██";AT 8,30;"█"
1020 PRINT INK 4;AT 6,0;"████"
1030 FOR n=8 TO 13: PRINT INK 6;
AT n,0;"+++": NEXT n
1040 PRINT AT 7,0; INK 7;"+++";A
T 14,0;",,,,";AT 13,16;"++++";AT
14,16;",++,"
1050 PRINT INK 3;AT 12,5;"###";A
T 12,18; INK 1;"█████"
1060 PRINT INK 5;AT 3,18;"C";AT
4,18;"C";AT 5,18;"C"
1100 REM CATAS FANTALLA
1102 RESTORE 1110
1105 FOR n=0 TO 29: READ PY,PX,T
,P,G: PRINT AT PY,PX; INK T; PAP
ER P;CHR$ G: NEXT N
1110 DATA 0,4,0,5,66,0,5,0,5,61,
1,7,0,5,61,2,8,5,0,35,2,9,0,5,65
,2,11,6,0,35,2,14,0,5,61
1120 DATA 1,16,7,5,35,0,19,6,5,3
5,1,21,7,5,33,2,21,7,5,34,3,20,0
,5,66,3,21,0,5,61,3,22,0,5,62
1130 DATA 2,23,0,5,64,2,24,0,5,6
5,2,26,0,5,145,3,27,0,5,145,4,28
,0,5,145,1,29,2,0,58,1,30,3,0,58
,1,31,2,0,58,1,27,7,5,35
1140 DATA 7,29,0,5,61,8,31,0,5,6
2,9,30,5,0,35,5,30,6,5,35,11,5,4
,0,38,1,9,7,5,35,3,29,6,5,35
1500 REM MOVIMIENTO
1501 IF HOJA=1 THEN PRINT ; INK
4;AT 10,5;"$";AT 9,5;"%";AT 8,5;
"$"
1510 PRINT INK 6;AT 6,18;"!";AT
7,18;CHR$ 34: FOR N=7 TO 10: PRI
NT INK 5; BRIGHT 1;AT N,18;"!";A
T N+1,18;CHR$ 34;AT N-1,18; INK
3;" ": GO SUB 7000: GO SUB 8000:
 GO SUB 7000: GO SUB 1517: NEXT
N
1515 PRINT INK 5; BRIGHT 1;AT 10
,18;"O";AT 11,18;"O": GO SUB 151
7: GO SUB 7000: GO SUB 8000: GO
SUB 7000: PRINT INK 0; BRIGHT 0;
AT 10,18;"█": GO SUB 1517: GO SU
B 7000: GO SUB 8000: GO SUB 7000
: PRINT INK 5; BRIGHT 1;AT 11,18
;"R": GO SUB 1517: GO SUB 7000:
GO SUB 8000: GO SUB 7000: PRINT
INK 0; BRIGHT 0;AT 11,18;"█"
1516 GO TO 1500
1518 IF Y=11 AND X=6 AND ESP=1 A
ND INKEY$="0" THEN LET FM=FM-1:
LET ESP=2: LET ANT=1: PRINT AT 1
9,6; PAPER 1;" ": LET HOJA=0: LE
T PORC=PORC+15: FOR Z=7 TO 0 STE
P -1: PRINT INK Z;AT 8,5;"$";AT
9,5;"%";AT 10,5;"$": NEXT Z
1520 RETURN
2000 REM PANTALLA N 2
2002 PRINT AT 15,0; INK 4; BRIGH
T 1;"::::::::::::::::::::::::::::
:::::"
2010 PRINT INK 2;AT 0,0;"::::::::
:::::::::::::::::::::::::::::::::
::::::::::::::::::::::::"
2020 PRINT INK 7;AT 3,26;", , ";A
T 12,13;"█████████";AT 11,15; INK 5
;"()"
2030 PRINT PAPER 4; INK 0;AT 9,5
;"B=@A"; INK 3; PAPER 0;AT 4,26;
"(')";AT 10,26;"(')"; INK 3;AT 0
,15;"  "
2040 IF BANT=0 THEN FOR N=5 TO 9
: PRINT INK 4;AT N,25;"+": NEXT
N
2099 RESTORE 2120
2100 FOR N=0 TO 18: READ PY,PX,T
,P,G: PRINT AT PY,PX; INK T; PAP
ER P;CHR$ G: NEXT N
2120 DATA 1,6,7,0,35,2,5,5,0,35,
2,13,5,0,35,6,0,7,5,35,5,5,1,0,4
2,2,26,6,0,43,2,28,6,0,43,2,27,7
,0,42
2130 DATA 4,1,6,5,35,2,0,7,5,33,
3,0,7,5,34,2,1,0,5,32,2,2,0,5,62
,3,1,0,5,32,4,0,0,5,32,5,1,0,5,3
2,5,0,0,5,32,6,1,0,5,144,7,0,0,5
,62
2500 REM MOVIMIENTO
2505 FOR N=133 TO 104 STEP -1: F
OR H=0 TO 2: GO SUB 8000: GO SUB
 2600: GO SUB 2700: PRINT INK RN
D*5+2;AT 9,15;"EE";AT 10,15;"EE"
: PLOT INK 0;220+H,N-1: PLOT INK
 7; OVER 1;220+H,N: NEXT H: NEXT
 N
2510 GO TO 2505
2602 IF X=15 AND Y=8 OR X=16 AND
 Y=8 THEN PRINT AT Y,X; INK 0;"
";AT Y-1,X; INK 0;" ": LET X=8:
LET Y=8: PRINT AT 8,8; INK 7;"<"
;AT 7,8;"J"
2605 IF X=5 AND Y=8 AND INKEY$="
0" AND ANT=2 AND BANT=0 THEN LET
 BANT=1: LET PORC=PORC+15: PRINT
 AT 4,5; INK 2; BRIGHT 1;"C": FO
R A=5 TO 9: PRINT AT A,25;" ": N
EXT A
2610 RETURN
2700 IF INKEY$="0" AND BANT=1 AN
D X=8 AND Y=8 AND FM>=0 THEN LET
 FM=FM-1: FOR F=9 TO 26 : PRINT
INK RND*5+2;AT 7,F;"G": PRINT IN
K 0;AT 7,F;"█": GO SUB 2800: NEX
T F
2750 RETURN
2800 IF N=115 AND F=26 THEN LET
GAR=1: PRINT AT 19,6; PAPER 1;"
": LET ANT=3: LET PORC=PORC+10
2810 RETURN
2999 PAUSE 0: GO SUB 6000: PAUSE
 0
3000 REM PANTALLA N 3
3010 PRINT INK 4;AT 0,15;"% $ $"
;AT 1,17;"% %";AT 2,19;"$";AT 15
,0; BRIGHT 1;"::::::::::::::::::::
:::::"; INK 1;"::::::::::::"
3020 PRINT INK 5;AT 0,22;"██████
██";AT 1,23;"█████████";AT 2,24
;"████████";AT 3,26;"██████": FO
R N=4 TO 8: PRINT INK 5;AT N,29;
"███": NEXT N
3030 RESTORE 3050
3040 FOR N=0 TO 16: READ PY,PX,T
,P,G: PRINT AT PY,PX; INK T; PAP
ER P;CHR$ G: NEXT N
3050 DATA 0,31,6,5,35,1,25,7,5,3
5,1,28,6,5,33,2,28,6,5,34,4,30,7
,5,33,5,30,7,5,34,9,30,7,0,33,10
,30,7,0,34
3060 DATA 8,29,0,5,61,5,28,0,5,6
1,4,28,0,5,32,4,27,0,5,61,4,26,0
,5,66,3,25,0,5,61,1,22,0,5,145
3070 DATA 2,23,5,0,35,3,24,4,0,3
7,4,24,4,0,36,8,31,0,5,64
3500 REM MOVIMIENTO
3520 LET CX=(RND*4)+6: FOR D=14
TO 11-(RND*4)-1 STEP -1: PRINT I
NK 5; BRIGHT 1;AT D,CX;"D": GO S
UB 7000: GO SUB 8000: GO SUB 700
0: PRINT INK 0;AT D,CX; BRIGHT 0
;" ": GO SUB 3600: NEXT D: GO TO
 3510
3600 IF Y=14 AND X=24 AND ANT=1
```





```
THEN LET ANT=2: LET PORC=PORC+15: PRINT AT 19,6; INK 3; PAPER 1;"."
3605 IF ANT=1 THEN PRINT INK 2; BRIGHT 1;AT 14,24;"."
3610 IF ANT=0 THEN PRINT INK (RND*5)+2;AT 5,24;"."
3620 RETURN
4000 REM  PANTALLA N 4
4005 PRINT INK 1; BRIGHT 1;AT 15,0;"::::::::::::::::::::::::::::::::::::"
4010 PRINT INK 5;AT 0,0;"████████";AT 2,0;"████████████";AT 3,0;"██████████";AT 4,0;"████████████";AT 5,0;"█████";AT 6,1;"██████"
4020 PRINT INK 4;AT 0,24;"A██████";AT 5,24;"A██████A"
4030 PRINT INK 7;AT 1,25;"++    +"; INK 5;AT 2,25;"++    +";AT 7,21;"B██A";AT 12,14;"█████";AT 10,18; INK 0; PAPER 5;"= @A"
4035 PRINT AT 8,22; INK 3;"???"
4040 RESTORE 4050
4045 FOR N=0 TO 44: READ PY,PX,T,P,G: PRINT AT PY,PX; INK T; PAPER P;CHR$ G: NEXT N
4050 DATA 3,25,5,0,43,3,26,5,0,43,4,25,7,0,44,4,26,7,0,44,3,28,3,0,77,3,29,3,0,78,4,28,3,0,79,4,29,3,0,80,4,30,7,0,44,3,30,5,0,43
4060 DATA 0,6,5,0,61,1,1,5,7,61,1,2,5,7,62,1,0,0,5,32,1,3,0,5,32,1,4,7,5,33,1,5,0,5,32,1,6,0,5,32,1,7,7,5,35,1,8,5,0,145
4070 DATA 2,4,7,5,34,3,1,6,5,35,5,4,5,7,66,5,5,5,7,61,5,6,5,7,62,5,7,0,5,32,5,8,0,5,144,6,0,4,5,33
4080 DATA 6,0,3,5,33,6,7,0,5,62,7,0,3,5,34,7,1,0,5,144,8,0,0,5,65
4090 DATA 7,5,4,0,58,7,6,4,0,58,8,5,4,0,58,8,6,4,0,58,9,5,2,0,40,9,6,2,0,41,15,5,2,0,40,15,6,2,0,41,9,22,3,0,40,9,23,3,0,41,15,22,3,0,40,15,23,3,0,41
4500 REM  MOVIMIENTO
4505 FOR N=11 TO 14: PRINT INK 2; BRIGHT 1;AT N,5;"GG";AT 25-N,22;"GG": GO SUB 7000: GO SUB 8000: GO SUB 4600: PRINT AT N,5; INK 0; BRIGHT 0;"  ";AT 25-N,22;"  ": NEXT N: GO TO 4500
4600 IF GAR=2 AND X=24 AND Y=4 AND INKEY$="0" THEN LET PORC=PORC+20: GO TO 9900
4605 RETURN
5000 REM  PANTALLA N 5
5005 PRINT AT 15,0; INK 4; PAPER 7;"????????????????????????????????????"
5007 PRINT AT 15,0; INK 1; PAPER 7;"::::"
5010 PRINT AT 14,12; INK 6;",,"; INK 5;",,"; INK 6;",,"; INK 1;"█▄█"; INK 7;",,"; INK 3;"██"; INK 6;",,"; INK 1;"█▄█"; INK 5;",,"; INK 4;"██"; INK 7;",,,,"
5020 PRINT INK 2;AT 13,15;"A█B"; INK 4;AT 13,22;"A█B"
5030 PRINT INK 4; PAPER 7;AT 0,3;"::::::::::::::::::::::::::::::"
5040 FOR N=2 TO 7: PRINT INK 6;AT N,4;"+": NEXT N
5045 FOR N=2 TO 5: PRINT INK 5;AT N,5;"+": NEXT N
5050 FOR N=10 TO 13: PRINT INK 7;AT N,29;"+++": NEXT N
5102 RESTORE 5150
5105 FOR N=0 TO 18: READ PY,PX,T,G: PRINT AT PY,PX; INK T; PAPER 0;CHR$ G: NEXT N
5150 DATA 1,4,7,42,1,5,3,42,1,6,7,42,2,6,6,43,3,6,6,43,1,9,5,43,1,10,7,43,1,11,6,43,2,9,5,43,1,19,4,36,1,21,4,37,2,21,4,37,7,31,7,43,8,31,7,43
5160 DATA 9,30,7,43,9,31,7,43,7,31,7,42,9,30,7,42,10,29,7,42
5500 REM  MOVIMIENTO
5505 IF ESP=0 THEN PRINT INK 5; BRIGHT 1;AT 12,16;"-"
5510 IF GAR=1 THEN PRINT INK 2; BRIGHT 1;AT 12,23;"/"
5550 IF GAR=0 THEN PRINT INK (RND*5)+2;AT 12,23;"G"
5560 IF X=16 AND Y=12 AND ESP=0 THEN LET ESP=1: LET PORC=PORC+10: PRINT AT 19,6; INK 7; PAPER 1;"-"
5570 IF X=23 AND Y=12 AND GAR<>2 THEN LET GAR=2: PRINT AT 19,6; INK 3; PAPER 1;"/": LET PORC=PORC+15
5580 GO SUB 8000
5600 GO TO 5550
5999 STOP
6000 REM  BORRADO
6010 FOR N=0 TO 15: PRINT INK 0;AT N,0;"                                ": NEXT N
6020 RETURN
7000 REM  CHOQUE
7010 IF ATTR (Y,X)=66 OR ATTR (Y-1,X)=66 OR ATTR (Y,X)=69 OR ATTR (Y-1,X)=69 THEN GO SUB 7030: FOR L=60 TO 0 STEP -2: PRINT AT Y,X; INK L/8;"<";AT Y-1,X;"J": OUT 254,RND*6+2: BORDER 0: NEXT L: LET Y=14: LET X=11: PRINT INK 7;AT Y,X;"<";AT Y-1,X;"J": INK 0
7020 RETURN
7030 IF ESCUDO=2 THEN LET FM=FM-1: RETURN
7040 IF ESCUDO<>2 THEN GO TO 9700
7500 REM  N.PANTALLA
7510 IF X=31 THEN LET PANT=PANT+1: LET PAST=PAST+1: LET X=1: PRINT INK 7;AT Y,X;"I";AT Y-1,X;"F"
7520 IF X=0 THEN LET PANT=PANT-1: LET PAST=PAST-1: LET X=30: PRINT INK 7;AT Y,X;"K";AT Y-1,X;"H"
7999 RETURN
8000 REM  RUTINA PRINCIPAL
8001 BRIGHT 0
8002 IF ATTR (Y+1,X)=0 THEN LET Y=Y+1: PRINT INK 7;AT Y,X;"<";AT Y-1,X;"J"; INK 0;AT Y-2,X;" ": GO TO 8001
8003 IF FM<=8 THEN PRINT AT 19,16; INK 4;"█";AT 19,11+FM/1.5;"█"
8004 IF FM=0 THEN GO TO 9700
8005 LET TIME=TIME-.2: PRINT AT 19,25; INK 2; BRIGHT 1;; PAPER 0;"  ";INT TIME;"  "
8006 IF TIME=0 THEN GO TO 9700
8010 IF INT (RND*1000)=88 AND ESCUDO<>2 THEN LET ESCUDO=1: PRINT AT 14,3; INK 5; BRIGHT 1;"D"
8020 IF X=3 AND Y=14 AND ESCUDO=1 THEN LET ESCUDO=2: PRINT AT 19,7; INK 6; PAPER 1;"D"
8100 IF INKEY$="5" AND (ATTR (Y,X-1)=0 OR ATTR (Y,X-1)=66 OR ATTR (Y,X-1)=69) THEN LET X=X-1: PRINT INK 7;AT Y,X;"K";AT Y-1,X;"H"; INK 0;AT Y,X+1;" ";AT Y-1,X+1;" "
8200 IF INKEY$="8" AND (ATTR (Y,X+1)=0 OR ATTR (Y,X+1)=66 OR ATTR (Y,X+1)=69) THEN LET X=X+1: PRINT INK 7;AT Y,X;"I";AT Y-1,X;"F"; INK 0;AT Y,X-1;" ";AT Y-1,X-1;" "
8300 IF INKEY$="7" AND (ATTR (Y-3,X+1)=0 OR ATTR (Y-3,X+1)=66 OR ATTR (Y-3,X+1)=69) AND (ATTR (Y-4,X+1)=0 OR ATTR (Y-4,X+1)=66 OR ATTR (Y-4,X+1)=69) THEN PRINT INK 0;AT Y,X;" ";AT Y-1,X;" ": LET Y=Y-3: LET X=X+1: PRINT INK 7;AT Y,X;"I";AT Y-1,X;"F"
8400 IF INKEY$="6" AND (ATTR (Y-3,X-1)=0 OR ATTR (Y-3,X-1)=66 OR ATTR (Y-3,X-1)=69) AND (ATTR (Y-4,X-1)=0 OR ATTR (Y-4,X-1)=66 OR ATTR (Y-4,X-1)=69) THEN PRINT INK 0;AT Y,X;" ";AT Y-1,X;" ": LET Y=Y-3: LET X=X-1: PRINT INK 7;AT Y,X;"K";AT Y-1,X;"H"
8449 IF X=31 OR X=0 THEN GO SUB 6000: GO SUB 7500: GO TO PANT*1000
8450 RETURN
9000 REM  GRAFICOS
9001 LET G=8
9005 RESTORE 9070
9008 PRINT INK 7; BRIGHT 1;AT 18,8;"GENERANDO U.D.G."
9010 FOR N=0 TO 399
9020 POKE 51464+N,PEEK (15616+N)
9030 NEXT N
9040 FOR N=51464 TO 51863
9050 READ A: POKE N,A: NEXT N
9060 RETURN
9070 DATA 255,249,115,118,118,50,50,52,52,52,60,28,24,24,24,8,255,255,183,183,178,178,176,32
9080 DATA 8,104,120,59,15,14,8,8,104,120,59,15,55,120,56,8,56,24,8,8,8,62,127,255
9090 DATA 255,0,255,0,255,0,255,0,255,120,255,120,255,120,255,0,255,30,255,30,255,30,255,0
9100 DATA 255,255,126,126,60,52,52,52,52,52,52,52,52,52,52,52,52,52,60,126,126,255,255
9110 DATA 9,6,6,9,16,32,64,128,56,111,214,252,248,136,144,224,3,7,14,60,120,248,240,240
9120 DATA 1,126,70,74,86,102,126,128,0,8,24,8,24,24,24,0,0,126,98,14,48,70,126,0
9130 DATA 0,124,68,28,6,70,126,0,0,12,22,38,126,12,12,0,0,62,32,62,6,38,62,0
9140 DATA 0,124,64,126,98,98,126,0,0,126,98,4,8,16,48,0,0,60,36,126,98,98,126,0,0,62,34,62,6,70,126,0
9150 DATA 191,191,191,0,247,247,247,0,24,126,60,0,247,247,247,0,102,36,255,189,195,126,0,90
9160 DATA 0,128,192,240,56,220,110,255,0,1,3,6,6,27,55,255,90,90,90,90,90,90,90,90,0,0,0,0,3,15,121,255
9170 DATA 3,7,63,254,254,241,231,255,192,224,248,126,127,153,252,255
9180 DATA 8,20,36,90,57,94,60,0,24,46,99,137,131,81,171,222,0,32,4,128,44,13,64,36
9190 DATA 0,28,60,58,250,222,152,135,0,0,0,170,85,0,0,0,0,56,60,92,95,123,25,225
9200 DATA 182,140,132,248,252,231,127,126,0,24,60,36,36,255,255,102,109,49,33,31,63,231,254,62
9210 DATA 0,8,12,254,255,254,12,8,0,1,7,15,1,0,0,63,0,128,128,128,192,224,112,248,31,0,63,63,0,127,127,0,248,12,252,252,6,254,254,0
9220 DATA 2,64,16,132,81,0,18,64,0,0,0,0,24,164,46,216
9500 REM  FINAL  DE LA AVENTURA
9690 GO TO 9800
9700 REM  MUERTO
9702 IF AV=0 THEN FOR N=7 TO 0 STEP -1: PRINT INK N;AT Y,X;"<";AT Y-1,X;"J": BEEP .2,20-N: NEXT N
9705 CLS : GO SUB 9800
9706 POKE 23607,60
9707 LET SC=INT (PORC*100+(1000-TIME))
9708 PRINT INK 7;AT 3,2;"D D LAS CUEVAS DE SNAKE D D"
9709 IF SC>HI THEN LET HI=SC
9710 PRINT INK 5;AT 6,5;"SCORE ";SC;AT 8,5; INK 4;"COMPLETADO ";PORC;" %";AT 10,5; INK 6;"HISCORE ";INT HI
9711 INK 3: BRIGHT 1: PLOT 30,70: DRAW 170,0: DRAW 0,12: DRAW -170,0: DRAW 0,-12: PLOT 30,69: DRAW 10,0: PLOT 35,83: DRAW 165,0: DRAW 0,-10
9712 PRINT INK 5;AT 19,4;"________________________"
9715 PRINT INK 7;AT 16,5;"RQG SOFWARE  "; INK 3;"© 1986"
9720 FOR M=1 TO 3: FOR N=0 TO 3: BEEP .15,30: BEEP .15,30+N*4: NEXT N: NEXT M
9721 IF PORC>=25 THEN LET B$="PORCENTAJE  REGULAR": IF PORC>=45 THEN LET B$=" PORCENTAJE  NORMAL": IF PORC>=70 THEN LET B$=" PORCENTAJE  BUENO"
9722 IF AV=1 THEN LET B$="AVENTURA COMPLETADA"
9723 FOR N=1 TO LEN B$: PRINT AT 12,5; INK 7; BRIGHT 1;B$( TO N): PAUSE 10: NEXT N
9725 PRINT INK 7;AT 3,2;"D D "; INK RND*6+2;"LAS CUEVAS DE SNAKE "; INK 7;"D D"
9730 IF INKEY$<>"" THEN GO TO 10: GO TO 9725
9749 GO TO 9725
9800 REM  BORDE
9807 PRINT INK 7;AT 0,0;"████████████████████████████████";AT 21,0;"████████████████████████████████"
9810 FOR N=0 TO 20: PRINT INK 7;AT N,0;"█";AT N,31;"█": NEXT N
9820 PRINT INK 7;AT 0,0;"B";AT 0,31;"B";AT 21,0;"█";AT 21,31;"█"
9830 RETURN
9900 REM  FINAL
9910 FOR A=56 TO 0 STEP -8
9915 RESTORE 9940
9920 FOR F=23300 TO 23300+15
9930 READ B: POKE F,B: NEXT F
9940 DATA 38,A,1,0,88,17,192,2,124,2,3,27,122,179,32,248,201
9950 RANDOMIZE USR 23300
9956 PRINT INK 7; PAPER A/8;AT 3,28;"EE";AT 4,28;"EE"
9960 NEXT A
9970 LET AV=1
9980 GO TO 9700
```

BOMB JACK. Arcade. Elite

EL SUPER-RATON VOLADOR

A simple vista parece que este diminuto ratón volador no es nada especial. Pero no debemos fiarnos de las apariencias puesto que Bom Jack es uno de los programas más adictivos y entretenidos de cuantos han hecho aparición durante los últimos meses.

¿Qué es aquello que vuela a toda velocidad entre los edificios? ¿Será el Halley?, ¿será un F18?, ¿será el IVA?

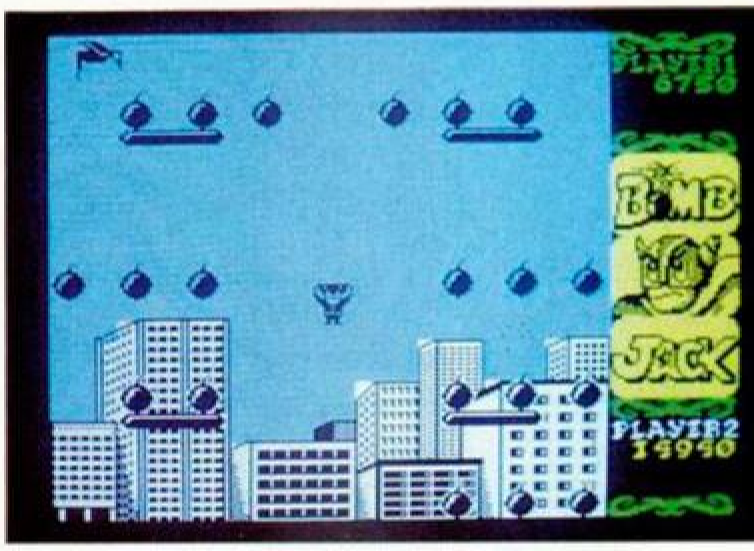
¡No!, ¡es Bomb Jack!, el ratón volador más rápido del mundo. Su destreza y habilidad le han convertido en la única esperanza de que dispone la Tierra para evitar ser destruida por una invasión de alienígenas llegados del espacio. El es el único ser que está en condiciones de desactivar todas las bombas que estos invasores han situado estratégicamente en los lugares más señalados del planeta.

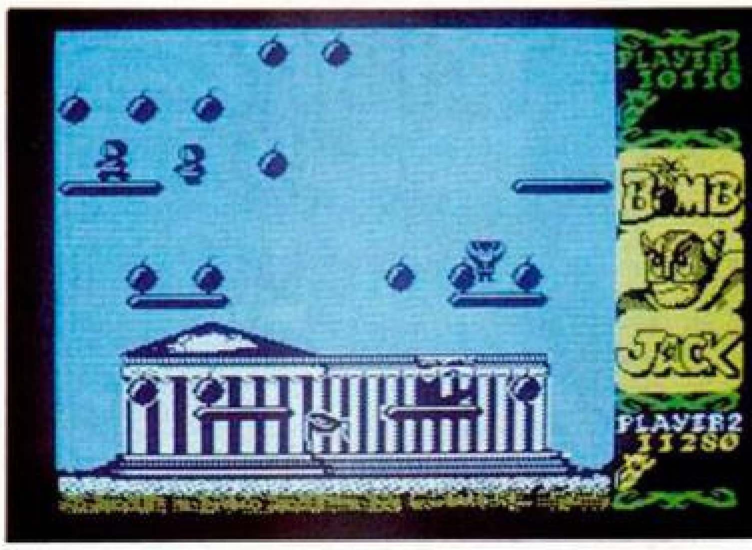
Pero no adelantemos acontecimientos y comencemos a explicar los antecedentes que rodean a este excelente programa llamado Bomb Jack.

Como ya sabréis los asiduos lectores de nuestra revista, y en especial aquéllos que sigan la sección de Micropanorama, este nuevo programa, realizado por Elite, está basado, o mejor dicho, es una muy fiel reproducción de uno de los juegos más famosos existentes actualmente en las máquinas de vídeo.

Esta experiencia ya había sido llevada a cabo anteriormente por Elite, pues como recordaréis, el archipopular Comando también tuvo su origen en dicho tipo de máquinas y en vista del éxito alcanzado, los directivos de la casa decidieron realizar una nueva tentativa que, a buen seguro, alcanzará unas cotas tanto o más altas que las de su predecesor.

Pero no creáis que esta afirmación la hacemos de una manera gratuita, ya que es de suponer que si Comando, con una calidad justita y con unos gráficos más pobres, llegó a ser lo que es, pues mucho más lógico será afirmar que Bomb Jack, que posee unos gráficos bastante vistosos (aunque escasos), una calidad sensiblemente superior y una adicción similar, puede llegar a desbancar no sólo a éste sino a cualquier otro programa que se interponga en su camino. Aunque también hay que reconocer que esto del software es completamente imprevisible y nunca se puede afirmar con seguridad lo que va a ocurrir con un programa. Así pues,







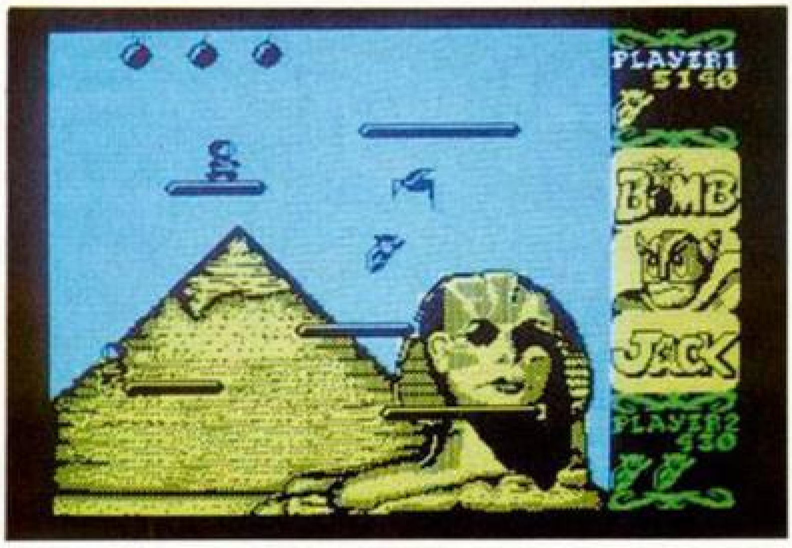
atendamos más directamente los asuntos propios de este Bomb Jack.

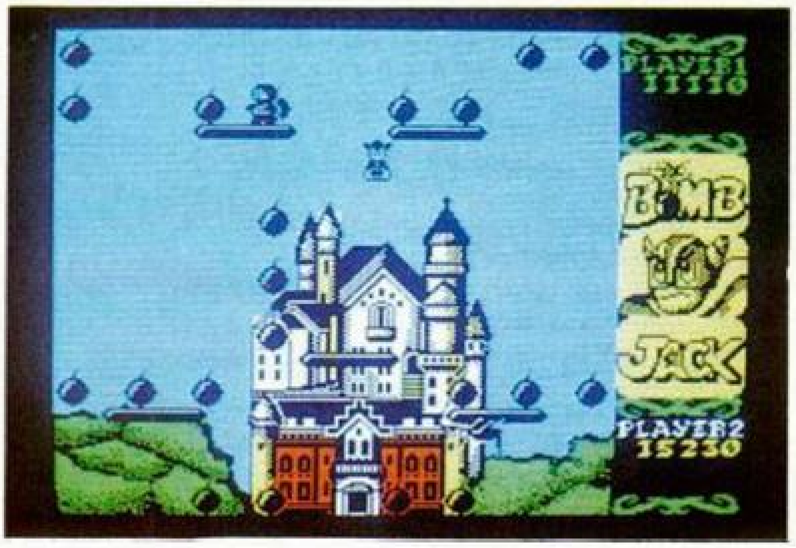
Como decíamos al principio, el protagonista de este simpático arcade es un diminuto ratonzuelo volador, hábil como el que más y veloz como pocos. Su misión consiste en pasearse por las construcciones más famosas de la Tierra: las Pirámides de Egipto, la Acrópolis de Atenas, los rascacielos de Nueva York y un opulento castillo centroeuropeo, e ir recogiendo en estos diferentes escenarios las innumerables bombas que los alienígenas de turno han colocado con vaya usted a saber qué mal intencionado fin.

Estos explosivos, por supuesto, se encuentran protegidos por sus mismos invasores y Jack deberá darse toda la prisa que pueda en recogerlos o de lo contrario irán apareciendo cada vez un mayor número de enemigos que intentarán dificultarle aún mucho más las cosas.

Jack no dispone de ningún tipo de armas para protegerse de los ataques contrarios, por lo que prácticamente lo único que puede hacer es esquivar a los seres flotantes y evitar ser rozado por cualquiera de ellos.

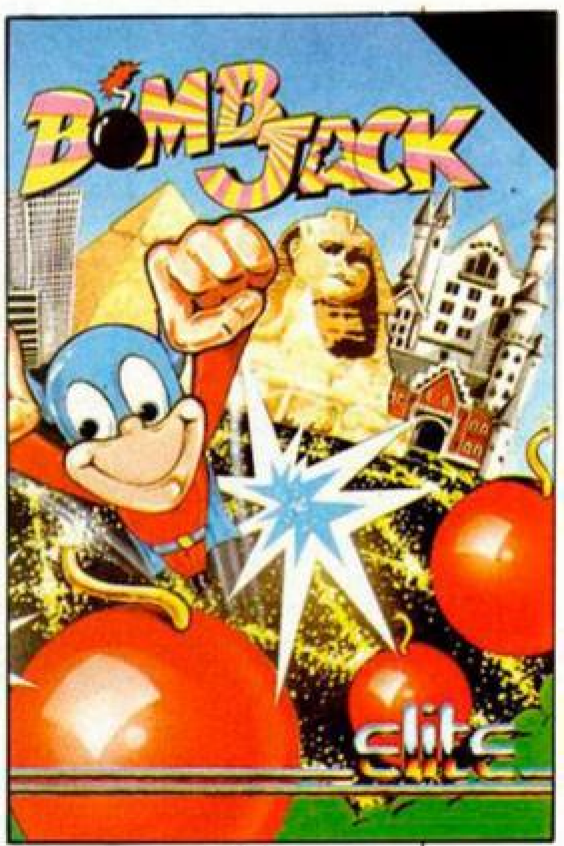
Sin embargo, afortunadamente para todos, cada cierto tiempo caen del cielo unos extraños paquetes que si logramos alcanzarlos harán que todos los alienígenas queden temporalmente paralizados, lo que nos permitirá, además de movernos con más libertad por los escenarios, el poder acabar con ellos simplemente por el hecho de tocarlos.



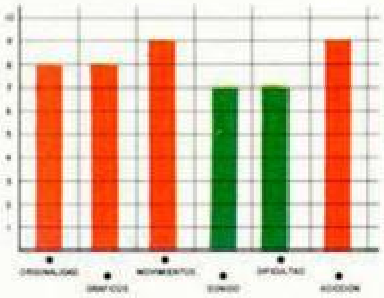


A medida que vayamos limpiando de bombas las distintas construcciones, el nivel de dificultad irá aumentando y cada vez resultará más y más entretenido el jugar con este Bomb Jack.

Como veis, el argumento del programa es bastante simple, ya que esto es básicamente todo lo que tenemos que hacer durante el transcurso del juego, pero a pesar de ello y gracias a que la colocación de las bombas va cambiando constantemente, el resultado final no resulta en absoluto monótono, sino que por el contrario, es de lo más divertido.



En definitiva, Bomb Jack es un programa que presenta un gran número de virtudes, tanto en sus aspectos gráficos como en los de divertimento, por lo que, sin duda alguna, dentro de poco tiempo oiremos hablar mucho de él.

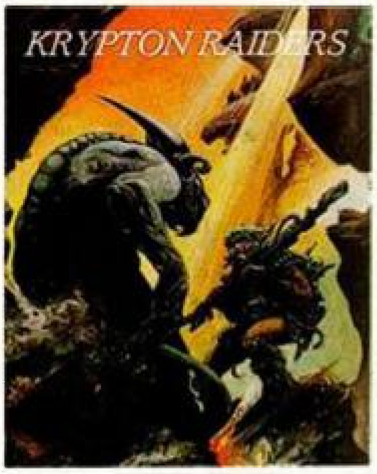




¡NUEVO!

KRIPTON RAIDERS • Arcade • Future Star

TRAS LOS DIAMANTES DE KRIPTON



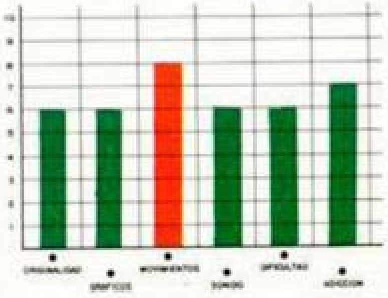
Kripton Raiders es otro de los nuevos programas que Dinamic ha lanzado al mercado bajo el sello de Future Star.

El programa consiste en un arqueotípo arcade laberíntico, en el cual nuestro héroe, ayudado por un autopropulsor que le permite moverse libremente en todas direcciones, debe intentar recoger cuatro enormes diamantes que se encuentran ocultos entre las cientos de galerías de la enrevesada mina de Kripton.

Pero la principal dificultad del juego reside en la gran cantidad de animalejos, insectos y demás extraños artefactos que pululan incansables y amenazadores por todo el interior de la mina. Afortunadamente para todos, nuestro pequeño amigo cuenta con una pistola lanzadora de rayos catatónicos con la cual puede defenderse de los continuos ataques por parte de sus enemigos; pero, por supuesto, deberá afinar al máximo la puntería y tendrá que ser verdaderamente rápido si quiere evitar ser destruido.

Kripton Raiders resulta un juego bastante entretenido y muy ameno, aunque quizás por el hecho de que tampoco presenta excesiva variedad en su desarrollo, cuando llevamos un rato largo jugando puede llegar a cansar ligeramente; aunque la verdad es que de primeras se coge con bastantes ganas y durante un tiempo resulta muy adictivo.

En lo referente a los aspectos gráficos del mismo, podemos decir que, sin llegar a poseer unas características muy destacadas, pues prácticamente todas las pantallas son casi iguales y tampoco posee unos dibujos excesivamente cuidados, el resultado general es aceptable, aunque también tenemos que tener en cuenta que se trata de la obra de un programador novel y para ser la primera vez, tampoco podemos esperar demasiado.







GLADIATOR • Deportivo • Domarc

EL CAMINO HACIA LA LIBERTAD

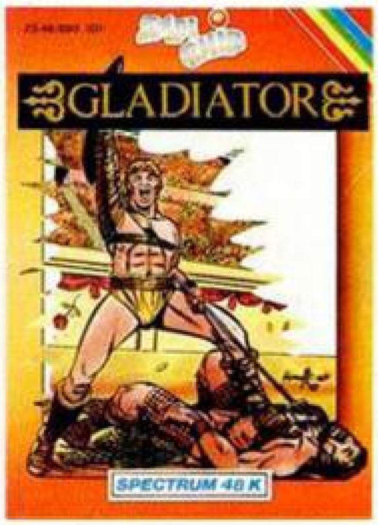
¡Libertad o muerte! Este era el grito con el que los gladiadores de la Antigua Roma comenzaban todos sus combates. Sabían que allí, en el centro de la arena, ante miles de espectadores ansiosos de sangre, sólo tenían dos opciones: matar o morir.

Sabían perfectamente que si querían conseguir su libertad, deberían vencer a todos sus adversarios con el fin de reunir la cantidad de dinero suficiente que les permitiera comprarla.

Esto fue lo que le ocurrió a Marcus de Messina, un hercúleo joven que fue apresado en su pacífica aldea, por las legiones romanas y entrenado en las mortíferas artes de los gladiadores.

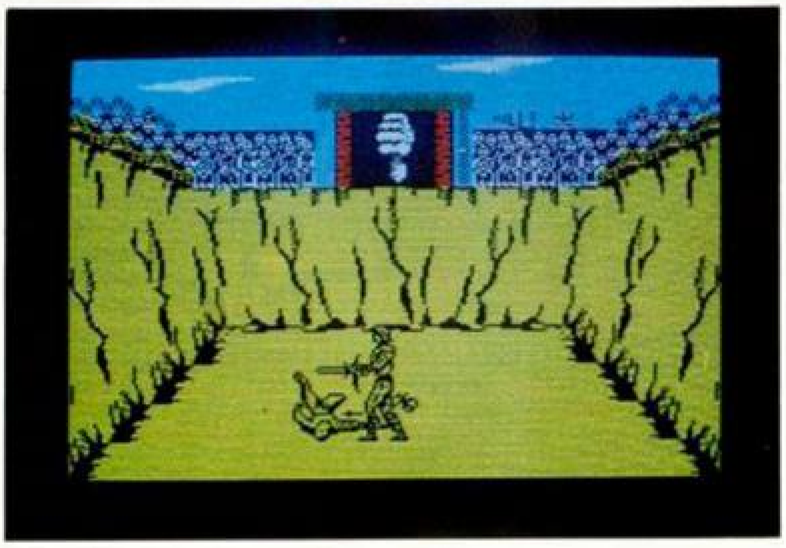
El Emperador sabe que es el mejor luchador, pero Marcus aún no tiene el dinero necesario para comprar su libertad.

Así pues, mucho tendrá que pelear aún. En cada combate tendrá que elegir tres de las 45 posibles armas. Con ellas podrá realizar diferentes tipos de combinaciones, adecuándose así a las características de lucha de sus oponentes y sacando un máximo partido de ellas.



Marcus deberá vencer, utilizando todas sus increíbles dotes de combatiente, a un mínimo de 14 gladiadores. Pero con estas victorias aún no podrá conseguir su libertad, ya que el dinero obtenido no será suficiente, por lo que después tendrá que apostar sobre sí mismo, aumentando así considerablemente su capital. Una vez que haya conseguido las 180 águilas que necesita, Marcus recibirá de propias manos del Emperador el Certificado de Libertad.

Y esto es en lo que consiste este estupendo Gladiator, un juego en el que

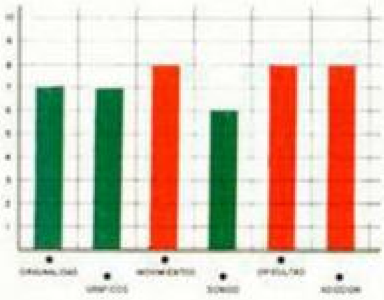






las posibilidades de lucha son enormes debido a la increíble variedad de movimientos diferentes que se pueden llevar a cabo, dependiendo de que no pulsemos el botón de fuego o lo hagamos una o dos veces, lo que supone un total de 24 opciones distintas. Esta circunstancia provoca una cierta confusión en los primeros combates ya que es bastante difícil conseguir llevar a cabo el movimiento que realmente deseamos, y lo más lógico es que el luchador realice cualquier acción menos la que nosotros queremos.

Los diseños de los gladiadores y los diferentes escenarios también están muy cuidados, con unos resultados muy atractivos y que además el juego está plagado de simpáticos pequeños detalles, no nos queda más que decir que Gladiator es un excelente programa que hará las delicias de todos, aunque también debemos advertir que al principio cuesta bastante el hacerse con el control de nuestro luchador y que sufriremos numerosas derrotas. Pero, ya sabéis... el camino hacia la libertad es realmente duro.



BENNY HILL • Arcade • Dk'Tronics

LAS AVENTURAS DE UN GORDINFLON

Parece que Dk'Tronics le ha cogido el gusto a esto de realizar programas para ordenador basándose en personajes de la televisión. Primero fue Popeye quien se vio en el trance de luchar denodadamente por el amor de Olivia, y en esta ocasión le ha tocado hacer de las suyas al simpático y gordinflón Benny Hill, conocido humorista británico que también alcanzó un notable éxito en nuestro país gracias a aquella serie televisiva que llevaba su propio nombre.

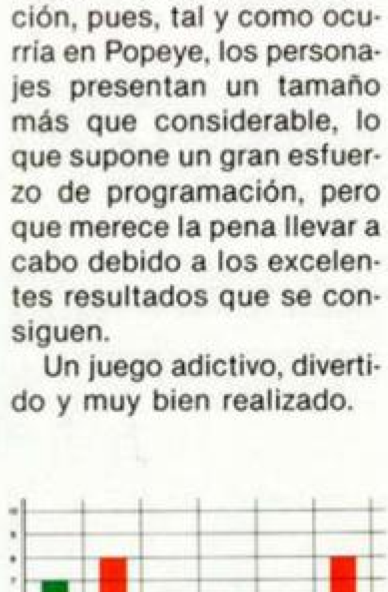
En este programa, de características muy similares a las de su predecesor, tanto en sus aspectos gráficos como en el desarrollo en sí mismo, representamos, ¡cómo no!, el papel del travieso Benny Hill y nos veremos envueltos en una serie de aventuras diferentes en las cuales tendremos que ir robando los respectivos objetos que vayan apareciendo en cada una de ellas (ropa, manzanas, etc.).

Estos objetos no estarán protegidos por innumerables peligros, sino que simplemente tendremos que evitar ser capturados por un único personaje en cada ocasión, quien nos perseguirá incansable por todas las pantallas. Además, también tendremos que eludir los numerosos obstáculos que se interpondrán en nuestro camino y que, a pesar de que no nos quitarán vidas, harán que perdamos parte de nuestro precioso tiempo para recoger cada uno de los objetos.

Este juego, a pesar de que en un principio parece muy simple, resulta bastante adictivo y, a medida que nos van quedando menos cosas por recoger y el tiempo se va haciendo más escaso, la emoción crece y al final se convierte en un auténtico «pique», y debido a que las aventuras son bastante diferentes entre sí, al pasar de una a otra, el ciclo de diversión se vuelve a repetir.

En cuanto a los gráficos de este Benny Hill, también llaman bastante la atención, pues, tal y como ocurría en Popeye, los personajes presentan un tamaño más que considerable, lo que supone un gran esfuerzo de programación, pero que merece la pena llevar a cabo debido a los excelentes resultados que se consiguen.

Un juego adictivo, divertido y muy bien realizado.



LOS JUSTICIEROS DEL SOFTWARE

WINTER SPORT

Electric Dreams, nos ofrece con este juego toda la espectacularidad de los deportes de invierno, una serie de pruebas en la nieve que nuestros justicieros han tenido ocasión de enjuiciar objetivamente, como es habitual ya en esta sección.

«Gráficos estupendos»

• POSITIVO

En general presenta buenos gráficos. Los jugadores de hockey son los personajes mejor realizados y tienen un movimiento muy gracioso. La cuarta y quinta prueba son muy adictivas aunque esta última es la más complicada.

• NEGATIVO

Falta melodía. Los tres primeros deportes son más bien regulares. El biathlon es un tanto complicado ya que apenas nos ayuda el mapa.

Puntuación: 8

Roberto Carlos Alonso

«Un juego deportivo más»

• POSITIVO

Esta muy bien el poder cargar las diferentes pruebas por separado. Los decorados de fondo están logrados. Los gráficos y movimientos son buenos. Algunas pruebas hacen del programa un juego adictivo, ya que están muy bien realizadas.

• NEGATIVO

Hay pruebas que por su lentitud llegan al aburrimiento y no están muy bien conseguidas. En cuanto a su originalidad es un juego deportivo más, eso sí con diferentes pruebas.

Puntuación: 6,5

David de Llodio

«Alta dificultad»

• POSITIVO

Esta bien presentado a nivel de menús e instrucciones pero se echa de menos una visión general de la pista en todas las pruebas. La dificultad es muy alta y es necesario plantear una estrategia antes de realizar una prueba.

• NEGATIVO

Los gráficos son muy sencillos y resultan poco vistosos. Se controla mal por el teclado. El movimiento en algunas pruebas, es lento y en cambio en otras demasiado rápido.

Puntuación: 6

Pilar Arias

«Los gráficos son pobres»

• POSITIVO

La pantalla de presentación está muy bien conseguida. Tiene una gran variedad de deportes, todos ellos bastante entretenidos. Las pruebas están muy de acuerdo con la realidad.

• NEGATIVO

Se echa en falta la variedad de sonido. Los gráficos son bastante pobres. En hockey el movimiento es lento.

Puntuación: 6

José Antonio Galiana

«La prueba de hockey, la mejor»

• POSITIVO

Es un juego en el que destaca la gran variedad de pruebas, lo que hace que sea bastante entretenido e interesante. Los gráficos, en general, son buenos y cabe destacar la prueba de hockey.

• NEGATIVO

El programa al tener tantas pruebas, algunas de éstas se hacen pesadas y aburridas sobre todo el slalon, lo cual quita vistosidad al juego.

Puntuación: 8

Juan Markessinis

«Escasez de efectos sonoros»

• POSITIVO

Las dos ventanas que salen de cada pantalla facilitan un mayor control del juego. La variedad de los deportes tratados, que lo hacen bastante entretenido y su originalidad, son aspectos bien conseguidos y realmente positivos.

• NEGATIVO

Los gráficos no están muy elaborados y las pantallas brillan por su ausencia. Se echa en falta algún efecto sonoro.

Puntuación: 6

Fco. Javier Cano

«El color, bien aplicado»

• POSITIVO

Los gráficos son bastante buenos en general. El color está bien aplicado. El movimiento es aceptable. Es bastante original por las pruebas que reúne lo cual hace que la adicción sea elevada. La dificultad en las pruebas es intermedia.

• NEGATIVO

El efecto de velocidad no está muy bien conseguido y hay pruebas que dejan mucho que desear. El sonido podría estar mejor realizado.

Puntuación: 7

Ricardo Alonso

«Presentación original»

• POSITIVO

La presentación es bastante original, también por la disposición de la pantalla en dos ventanas, viendo así el juego en dos posiciones distintas. El movimiento y el efecto de fondo están muy bien logrados. Los más adictivos son el hockey y el biathlon. Los gráficos son aceptables aunque no llegan a ser excelentes.

• NEGATIVO

El sonido es escaso y malo. La mayoría de las pruebas son muy lentas. El slalon se hace largo y bastante aburrido y el patinaje de velocidad es lo mejor para romper teclados.

Puntuación: 7

Gabriel Martí Fuentes



```
601781190     INC   HL          Pass 2 errors: 00
601791200     LD    (HL),D
601801210     RET               Table used:   251  from   277
```

Fig. 9-20. Listado completo de la rutina de salida «Canal».

EJERCICIOS

1.- Escriba una rutina que sustituya a "IMPR_1" (puede llamarla "IMPR_2") y que imprima los caracteres en "imagen de espejo".

2.- Modifique la rutina "ACT" para que "IMP_A" trabaje por el canal "S".

3.- ¿Como haría para posicionar la impresión en un lugar determinado de la pantalla ayudándose de la rutina "IMP_A"?.

4.- Si ejecutamos el comando OPEN #4,"S" ¿que modificación se produciría en la tabla de corrientes?, ¿y en la de canales?.

5.- Escriba una rutina que multiplique por 32 el dato contenido en el acumulador, devolviendo el resultado en el registro "HL". Efectue la multiplicación mediante desplazamientos.

daremos una pista: «CALL SCROLL» se ensambla como «CD,FE,0D» y «CALL IMP-A» como «CD,60,EA». Para las direcciones se puede usar la siguiente tabla de equivalencia:

```
23606 = 5C36h
23684 = 5C84h
23681 = 5C81h
23688 = 5C88h
```

Vamos a ir viendo el ensamblado por trozos tal y como vimos las rutinas:

```
100  60000  ED,5B,36,5C
110  60004  26,00
120  60006  6F
130  60007  29
140  60008  29
150  60009  29
160  60010  19
170  60011  EB
180  60012  2A,84,5C
```

Esta es la parte donde calculábamos las direcciones en el font y en la pantalla. A continuación viene la rutina «IMPR-1»:

```
190  60015  06,08
200  60017  1A
210  60018  77
220  60019  3A,81,5C
230  60022  E6,01
240  60024  28,0F
250  60026  CB,3E
260  60028  78
270  60029  FE,05
280  60031  30,08
290  60033  CB,26
300  60035  FE,03
310  60037  30,02
320  60039  CB,26
330  60041  3A,81,5C
340  60044  E6,02
350  60046  28,05
360  60048  7E
370  60049  CB,3F
380  60051  B6
390  60052  77
400  60053  13
410  60054  24
420  60055  10,DB
```

A continuación viene la parte encargada de actualizar las coordenadas:

```
440  60057  ED,5B,88,5C
450  60061  21,21,18
460  60064  ED,52
470  60066  EB
480  60067  1C
490  60068  7B
500  60069  FE,20
510  60071  38,0E
520  60073  1E,00
530  60075  14
540  60076  7A
550  60077  FE,15
560  60079  38,06
570  60081  CD,FE,0D
580  60084  11,00,14
```

Finalmente, la parte encargada de calcular la nueva dirección de pantalla y almacenar ésta y las coordenadas en las variables del Sistema correspondiente:

```
590  60087  D5
600  60088  7A
610  60089  E6,07
620  60091  CB,0F
630  60093  CB,0F
640  60095  CB,0F
650  60097  B3
660  60098  5F
670  60099  7A
680  60100  E6,18
690  60102  F6,40
700  60104  57
720  60105  ED,53,84,5C
730  60109  D1
740  60110  21,21,18
750  60113  ED,52
760  60115  22,88,5C
770  60118  C9
```

Con esto, queda completa la rutina «IMP-A». Ahora vamos a ensamblar la rutina «TEST»:

```
1010  60500  21,65,EC
1020  60503  06,14
1030  60505  7E
1040  60506  C5
1050  60507  E5
1060  60508  CD,60,EA
1070  60511  E1
1080  60512  23
1090  60513  C1
1100  60514  10,F5
1110  60516  C9
1120  60517  63,75,72,73
             6F,20,43,2F
             40,20
1130  60527  4D,49,43,52
             4F,48,4F,42
             42,59
```

PROGRAMA 9-1

```
  10 REM    PROGRAMA 9-1
------------------------------
  15 REM  CARGA CODIGO MAQUINA
  20 DEF FN a(a$,n)=16*(CODE a$(
n)-48-7*(a$(n)>"9"))+(CODE a$(n+
1)-48-7*(a$(n+1)>"9"))
  25 CLEAR 59999
  30 FOR h=1 TO 2: READ d,l
  40 FOR f=1 TO l: READ a$,s: LE
T c=0
  50 FOR n=1 TO 19 STEP 2
  60 LET a=FN a(a$,n): POKE d,a:
 LET d=d+1: LET c=c+a: NEXT n
  70 IF c<>s THEN PRINT "(ERROR)
 en la linea: ";1000+10*f+120*(l
=2): STOP
  80 NEXT f: NEXT h
 100 REM  DEMOSTRACION
 105 POKE 23681,3: PRINT AT 0,5;
: RANDOMIZE USR 60500
 110 INPUT "Coordenadas:";li,co
 120 INPUT "Texto: ";a$

 130 POKE 23658,0: POKE 23681,0
 140 INPUT "Cursiva ? (s/n)";r$:
 IF r$="s" THEN POKE 23681,1
 150 INPUT "Negrita ? (s/n)";r$:
 IF r$="s" THEN POKE 23681,PEEK
23681+2
 160 FOR n=1 TO LEN a$: POKE 605
16+n,CODE a$(n): NEXT n: POKE 60
504,LEN a$
 170 PRINT AT li,co;: RANDOMIZE
USR 60500: GO TO 110
1000 REM  IMP_A
1005 DATA 60000,12
1010 DATA "ED5B365C26006F292929"
,746
1020 DATA "19EB2A845C06081A773A"
,743
1030 DATA "815CE601280FCB3E78FE"
,1146
1040 DATA "053008CB26FE033002CB"
,812
1050 DATA "263A815CE60228057ECB"

,923
1060 DATA "3FB677132410D8ED5B88"
,1115
1070 DATA "5C212118ED52EB1C7BFE"
,1141
1080 DATA "20380E1E00147AFE1538"
,605
1090 DATA "06CDFE0D110014D57AE6"
,1080
1100 DATA "07CB0FCB0FCB0FB35F7A"
,1057
1110 DATA "E618F64057ED53845CD1"
,1404
1120 DATA "212118ED5222885CC900"
,872
1124 REM  TEST
1126 DATA 60500,4
1130 DATA "2165EC06147EC5E5CD60"
,1249
1140 DATA "EAE123C110F5C9637572"
,1479
1150 DATA "736F20432F4D204D4943"
,698
1160 DATA "524F484F424259000000"
,533
```

Esperamos que haya intentado ensamblar por sí mismo al menos esta última. Vamos a coger la rutina «IMP-A» y agrupar los bytes del código máquina de 10 en 10 formando 12 líneas de 20 caracteres (en la última línea faltarán dos caracteres pero rellenamos con «00»). A la izquierda de cada línea ponemos un número del 1 al 12 con lo que quedan numeradas. A la derecha, ponemos el resultado de sumar todos los octetos de la línea, pero expresado en decimal. Obtendremos algo así:

```
1 ED5B365C26006F292929 746
2 19EB2A845C06081A773A 743
3 815CE601280FCB3E78FE 1146
4 053008CB26FE033002CB 812
5 263A815CE60228057ECB 923
6 3FB677132410D8ED5B88 1115
7 5C212118ED52EB1C7BFE 1141
8 20380E1E00147AFE1538 605
9 06CDFE0D110014D57AE6 1080
10 07CB0FCB0FCB0FB35F7A 1057
11 E618F64057ED53845CD1 1404
12 212118ED5222885CC900 872
```

A continuación, hacemos lo mismo con «TEST»:

```
1 2165EC06147EC5E5CD60 1249
2 EAE123C110F5C9637572 1479
3 736F20432F4D204D4943 698
4 524F484F424259000000 533
```

Seguro que a la mayoría de los lectores les resulta familiar. Efectivamente, se trata del formato utilizado por el «Cargador universal de código máquina» publicado en MICROHOBBY. Evidentemente, resulta muy trabajoso calcular a mano las sumas de control de cada línea, pero no es difícil escribir un programa, en Basic, que lo haga. La ventaja de este formato es que, si se produce un error, se sabe exactamente en qué línea se ha producido y sólo hay que comprobar 20 caracteres para encontrarlo.

El Programa 9-1 utiliza este formato para cargar las rutinas «IMP-A» y «TEST» que nos permitirán escribir un mensaje de hasta 255 caracteres en cualquier lugar de la pantalla y utilizando letra cursiva o negrita. Si analiza el funcionamiento del programa a partir de la línea 100, verá la forma de utilizar estas rutinas en sus propios programas. Para salvarlas puede utilizar:

```
SAVE "IMP_A"CODE 60000,119
SAVE "TEST"CODE 60500,37
```

La rutina «IMP-A» es reubicable, la rutina «TEST» no y, además, no funcionará si «IMP-A» no está en la dirección 60000. Aunque, a estas

«AT», «TAB», «INK», etc. Puede utilizar este canal con «LPRINT» o con «PRINT #3» que es lo mismo. También puede hacer:

```
PRINT AT 10,12;#3;"HOLA"
```

Con lo que los códigos de control irán por el canal «S» y la palabra a imprimir irá por el «P», es decir, por el nuestro.

```
*HISOFT GENS3M ASSEMBLER*
      ZX SPECTRUM

  Copyright HISOFT 1983
  CURSO C/M MICROHOBBY

Pass 1 errors: 00

        10 *C-
        20 *D+
60000   90        ORG   60000
60000  100 CANAL  CP    13
60002  110        JR    Z,ENTER
60004  120        CP    32
60006  130        JR    NC,LB1
60008  140        RET
60009  150 LB1    CP    128
60011  160        JR    C,IMP_A
60013  170        RET
60014  180 ENTER  LD    DE,(S_POSN)
60018  190        LD    HL,#1821
60021  200        SBC   HL,DE
60023  210        EX    DE,HL
60024  220        JR    INC_LI
60026  230 IMP_A  LD    DE,(CHARS)
60030  240        LD    H,0
60032  250        LD    L,A
60033  260        ADD   HL,HL
60034  270        ADD   HL,HL
60035  280        ADD   HL,HL
60036  290        ADD   HL,DE
60037  300        EX    DE,HL
60038  310        LD    HL,(DF_CC)
60041  320 IMPR_1 LD    B,8
60043  330 BUC_1  LD    A,(DE)
60044  340        LD    (HL),A
60045  350        LD    A,(BAND)
60048  360        AND   1
60050  370        JR    Z,NOCURS
60052  380        SRL   (HL)
60054  390        LD    A,B
60055  400        CP    5
60057  410        JR    NC,NOCURS
60059  420        SLA   (HL)
60061  430        CP    3
60063  440        JR    NC,NOCURS
60065  450        SLA   (HL)
60067  460 NOCURS LD    A,(BAND)
60070  470        AND   2
60072  480        JR    Z,NOBOLD
60074  490        LD    A,(HL)
60075  500        SRL   A
60077  510        OR    (HL)
60078  520        LD    (HL),A
60079  530 NOBOLD INC   DE
60080  540        INC   H
60081  550        DJNZ  BUC_1
       560 ;
60083  570        LD    DE,(S_POSN)
60087  580        LD    HL,#1821
60090  590        SBC   HL,DE
60092  600        EX    DE,HL
60093  610        INC   E
60094  620        LD    A,E
60095  630        CP    32
60097  640        JR    C,SIGUE
60099  650 INC_LI LD    E,0
60101  660        INC   D
60102  670        LD    A,D
60103  680        CP    21
60105  690        JR    C,SIGUE
60107  700        CALL  SCROLL
60110  710        LD    DE,#1400
60113  720 SIGUE  PUSH  DE
60114  730 DIR    LD    A,D
60115  740        AND   #07
60117  750        RRC   A
60119  760        RRC   A
60121  770        RRC   A
60123  780        OR    E
60124  790        LD    E,A
60125  800        LD    A,D
60126  810        AND   #18
60128  820        OR    #40
60130  830        LD    D,A
       840 ;
60131  850        LD    (DF_CC),DE
60135  860        POP   DE
60136  870        LD    HL,#1821
60139  880        SBC   HL,DE
60141  890        LD    (S_POSN),HL
60144  900        RET
23606  910 CHARS  EQU   23606
23684  920 DF_CC  EQU   23684
23681  930 BAND   EQU   23681
23688  940 S_POSN EQU   23688
 3582  950 SCROLL EQU   #0DFE
60145 1000 ACT    LD    HL,(CHANS)
60148 1010        LD    DE,15
60151 1020        ADD   HL,DE
60152 1030        LD    E,(HL)
60153 1040        INC   HL
60154 1050        LD    D,(HL)
60155 1060        LD    (NMI),DE
60159 1070        LD    DE,CANAL
60162 1080        LD    (HL),D
60163 1090        DEC   HL
60164 1100        LD    (HL),E
60165 1110        RET
23728 1120 NMI    EQU   23728
23631 1130 CHANS  EQU   23631
60166 1140 DESACT LD    HL,(CHANS)
60169 1150        LD    DE,15
60172 1160        ADD   HL,DE
60173 1170        LD    DE,(NMI)
60177 1180        LD    (HL),E
```

Más adelante, podremos completar la rutina para que sea capaz de aceptar y manejar controles de posicionamiento como «AT» o «TAB». De momento, no deberemos mandar estos códigos, ya que producirían la impresión de cosas sin sentido.

Sin embargo, sí será necesario que la rutina sea capaz de aceptar el código «13» ya que el Basic siempre lo manda, a menos que la sentencia acabe en «;».

Lo mejor es construir una pequeña rutina «filtro» que sólo deje pasar códigos imprimibles, es decir, aquellos comprendidos entre 32 y 127 ambos inclusive. No obstante, este «filtro» debe ser capaz de identificar el código «13» para saltar a una rutina que lo gestione. Vamos a ir construyendo este filtro:

```
100 CANAL  CP   13
110        JR   Z,ENTER
120        CP   32
130        JR   NC,LB1
140        RET
150 LB1    CP   128
160        JR   C,IMP_A
170        RET
180 ENTER  LD   DE,(S_POSN)
190        LD   HL,#1821
200        SBC  HL,DE
210        EX   DE,HL
220        JR   INC_LI
```

Recuerde que entramos en esta rutina con el código en el acumulador. Así que lo primero que hacemos es compararlo con «13», si resulta que el código es 13, saltaremos a «ENTER» desde donde se llevará a cabo el incremento de línea. Exactamente, lo que hacemos es cargar las coordenadas en curso en «DE» y saltar a la etiqueta «INC-LI» (INCrementar LInea). Esta etiqueta deberemos colocarla en la línea 520 de la rutina «IMP-A» ya que a partir de este punto, lo que se hace es, precisamente, incrementar el número de línea.

Si el código no fuera «13», entraríamos en la línea 120 donde lo comparamos con «31» y retornamos si es igual o menor. Si no, saltamos a la línea 150 donde lo volvemos a comparar con «128» y retornamos si es igual o mayor. Finalmente, si el código está dentro del rango, saltamos a «IMP-A» para que se imprima.

Esta rutina de «filtro», deberá ir colocada inmediatamente antes de «IMP-A» de forma que podemos añadirla y ensamblarlas juntas. Como «IMP-A» es reubicable, no hay problema en cambiarla de dirección. En la Figura 9-20 podemos ver el listado completo de «CANAL» + «IMP-A» tal como quedaría al ponerlas juntas. Dejamos al lector la tarea de ensamblar a mano (si no tiene ensamblador, esta corta rutina.

Vamos a ver cómo «activamos» este canal. Ya que necesitamos los canales «S» y «K» para manejarnos por el Sistema, lo que haremos será colocar esta rutina en el canal «P», es decir, el correspondiente a la impresora. Para ello, no hay más que almacenar en «(CHANS) + 15» la dirección a partir de la cual hayamos ensamblado la rutina.

Mientras tengamos activado este canal, no podremos utilizar la impresora, así que será mejor que preveamos una forma de «desactivarla» y dejar el canal «P» con su anterior contenido. Almacenaremos éste de forma temporal en la dirección 23728 correspondiente a una variable que el Sistema no utiliza (El famoso vector de interrupción no enmascarable, hábilmente anulada por los simpáticos muchachos de Sinclair). Las rutinas de activar y desactivar pueden ser algo así:

```
1000 ACT    LD   HL,(CHANS)
1010        LD   DE,15
1020        ADD  HL,DE
1030        LD   E,(HL)
1040        INC  HL
1050        LD   D,(HL)
1060        LD   (NMI),DE
1070        LD   DE,CANAL
1080        LD   (HL),D
1090        DEC  HL
1100        LD   (HL),E
1110        RET
1120 NMI    EQU  23728
1130 CHANS  EQU  23631
1140 DESACT LD   HL,(CHANS)
1150        LD   DE,15
1160        ADD  HL,DE
1170        LD   DE,(NMI)
1180        LD   (HL),E
1190        INC  HL
1200        LD   (HL),D
1210        RET
```

Para activar podemos hacer un «RANDOMIZE USR» a la dirección donde está la etiqueta «ACT» y para desactivar, lo mismo pero a la dirección donde está «DESACT» (Si ensambla a partir de 60000, «ACT» estará en 60145 y «DESACT» en 60166).

Pruébelo y verá qué bien funciona. No olvide que puede controlar los flags de cursiva y negrita «POKEando» en la dirección 23681 y tenga en cuenta que no puede mandar códigos de control como

alturas, quien haya seguido atentamente el curso tiene que ser capaz de realizar las modificaciones necesarias para que puedan correr en cualquier otra dirección.

Ya hemos dicho que esta rutina podría ser una subrutina de un programa para gestionar la pantalla; de hecho, hay otra forma más fácil de utilizarla desde el Basic. Cualquier rutina en la que se entre con el código de un carácter en «A» y haga algo con ese carácter, puede ser utilizada por el Sistema Operativo como un canal de salida. Exactamente igual que el canal «S» (parte superior de la pantalla), el «K» (parte inferior) o el «P» (impresora). Para ello, vamos a ver qué son y cómo se utilizan los «Canales de comunicación».

Los canales de comunicación

La configuración básica de un ordenador está compuesta por el microprocesador y la memoria. No obstante, este conjunto no sirve para nada si no se puede comunicar con el exterior. Por tanto, lo normal es que se acompañe de un teclado, una pantalla y una impresora. Estos elementos se denominan periféricos.

A nivel máquina, sabemos que el microprocesador utiliza «ports» para comunicarse con todos los periféricos excepto la pantalla que está configurada como una «tabla» en memoria que la ULA se encarga de enviar al televisor. Para el microprocesador, esta tabla es como un periférico más y como tal es gestionada por el Sistema Operativo.

Cuando se trabaja en un lenguaje de alto nivel como el Basic, no resulta cómodo andar escribiendo y leyendo en «ports» o en posiciones de memoria. Es más fácil tener una o varias instrucciones de entrada y otras de salida y utilizarlas dirigidas a determinados canales que comuniquen con cada periférico. Pero ¿qué es un canal?

Existen canales de entrada y canales de salida. De momento, vamos a estudiar sólo los de salida porque son los que nos interesan para hacer funcionar nuestra rutina.

Un canal es, básicamente, una rutina escrita en código máquina que es llamada por el sistema operativo con el código de un carácter en «A». La rutina deberá enviar el carácter o el código al periférico correspondiente y, luego, devolverá el control al Sistema Operativo.

Un canal tiene que ser también capaz de manejar ciertos códigos de control tales como el retorno de carro (13), salto de línea (10), borrado (12) y, en general, aquéllos con códigos comprendidos entre «0» y «31» que incluyen controles de color, posicionamiento de la impresión, etc. Un mismo código puede tener significados distintos según el canal que se esté utilizando; por ejemplo, la impresora no aceptará controles de color.

El Spectrum, en su versión básica, utiliza cuatro canales. Se denominan «R», «S», «K» y «P». Algunos son sólo de salida y otros admiten también entradas. Suponemos que todos nuestros lectores están acostumbrados a utilizar los canales de comunicación en Basic, así que no profundizaremos más en ello y nos dedicaremos a estudiarlos desde el punto de vista del código máquina. Para mayor información sobre el cometido de cada canal, se puede consultar el manual de Basic que viene con el ordenador o el capítulo 29 del «CURSO DE BASIC» publicado por MICROHOBBY.

Sabemos que el Basic utiliza «corrientes» que enlazan con determinados canales. Algunas están ya asignadas como por ejemplo, las #1, #2 y #3 que enlazan, respectivamente, con los canales «K», «S» y «P». Toda la gestión de canales y corrientes es llevada a cabo por dos tablas de «ofset» que se encuentran en determinado lugar de la memoria. La primera es la tabla de corrientes situada a partir de la dirección 23568 y compuesta por 38 bytes. Cada elemento de esta tabla ocupa dos octetos que contienen (en el orden inverso habitual) un número que indica el «ofset» de la tabla de canales, es decir, el desplazamiento desde «CHANS» para acceder al canal unido a esa corriente.

Si la corriente en cuestión está cerrada, el elemento correspondiente de la tabla de corrientes contendrá «0000», justo como era de esperar. Todo esto se comprenderá mejor con una mirada a la Figura 9-19 donde hemos representado esquemáticamente las dos tablas de corrientes y canales, así como la conexión entre ellas.

En el momento de conectar el ordenador la situación es la que se muestra en la figura. Las corrientes #-3, #-2 y #-1 no son accesibles desde el Basic y sólo las utiliza el Sistema. De hecho, el Basic no nos admite la instrucción: OPEN #4, «R» ya que el canal «R» que conecta con el área de traba-

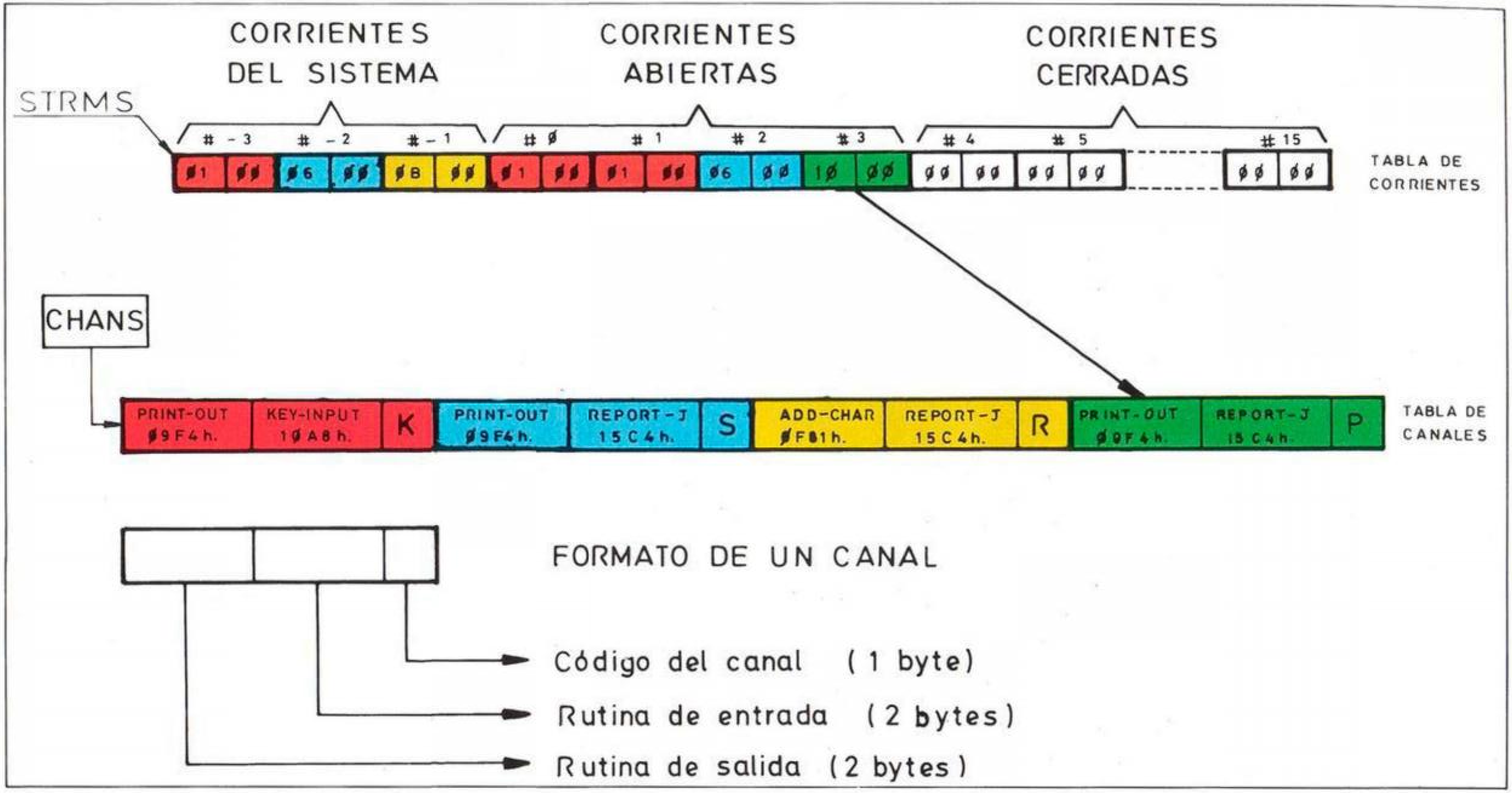


Fig. 9-19. Tablas de corrientes y canales.

jo sólo puede ser utilizado por el Sistema.

Las corrientes #Ø, #1, #2 y #3 están permanentemente abiertas y conectadas con sus respectivos canales. No hay inconveniente en utilizar la instrucción «OPEN #» para conectarlas con otros, pero si intentamos cerrarlas mediante «CLOSE #», obtendremos, por defecto, los canales inicialmente conectados. Habría, no obstante, una forma de cerrarlas que sería «POKEar» directamente «Ø» en «STRMS» + 1Ø con lo que quedaría cerrada la corriente #2. No obstante, no le aconsejamos que lo haga ya que el Sistema se «colgaría» con toda seguridad.

Conociendo la disposición de estas tablas, podemos hacer algo mucho más interesante. Podemos cambiar la dirección de salida de uno de los canales para apuntar a una rutina nuestra que gestione esta salida. Y aquí es donde entra «IMP-A»; esta rutina puede ser usada, con ciertas precauciones, como un canal de salida.

Vamos a ver qué nos manda el Basic a un canal. Supongamos que vamos a ejecutar la instrucción:

La salida se hará por la corriente #2 que es la que corresponde, por defecto, al comando «PRINT». Los códigos enviados por el Basic serán:

```
PRINT AT 10.12;"HOLA"
```

22-Código de control «AT».
1Ø-Primer argumento de «AT».
12-Segundo argumento de «AT».
72-Código de «H».
79-Código de «O».
76-Código de «L».
65-Código de «A».
13-Retorno de carro.

Todos los programas disponibles para el 128K

DAME MAS BYTES Y TE DARE MAS SOFTWARE

Sinclair no ha querido tropezar dos veces en la misma piedra, y los errores cometidos en el lanzamiento de algunos modelos anteriores han sido perfectamente rectificados para la presentación de su nuevo Spectrum 128.



De todos es conocido el hecho de que excelentes ordenadores como el QL, han supuesto un relativo fracaso comercial tan sólo por no haber aparecido en el mercado arropados por una software abundante en cantidad y calidad.

Por esta razón, Sinclair se aseguró, incluso antes de poner en práctica el proyecto del S 128, de que las casas productoras de programas en Gran Bretaña estuvieran decididas a apoyar firmemente su nuevo lanzamiento en la medida de sus posibilidades.

Y efectivamente, así ha ocurrido, puesto que prácticamente no ha quedado ni una sola compañía sin realizar, si no un programa específico para el 128, al menos sí una versión renovada de anteriores éxitos del S 48.

Este apoyo masivo ha sido debido, aparte de a las mayores posibilidades tanto de programación como de sonido que presenta el 128, a que entre todos se han propuesto crear una auténtica supermáquina de juegos, haciendo así un digno y mejorado sucesor del popular Spectrum 48.

De esta forma, tan sólo tras unos meses de lanzamiento oficial en Gran Bretaña, ya existe una nutrida lista de programas, tanto de divertimiento como de utilidades, que representan una extensa gama de posibilidades para el usuario.

A continuación os presentamos dos listas diferentes en las cuales se incluye la práctica totalidad del software actualmente disponible para el 128. El motivo de haber confeccionado dos listas obedece al hecho de que en la primera se presentan los programas de nueva realización, mientras que la segunda está formada por aquellos títulos que suponen una versión, aunque sea ligeramente diferente (introduciendo modificaciones tanto en las pantallas y diseños como en el sonido), de programas ya conocidos.

Novedades

Icon Graphix (Audiogenic Soft.): un poderoso diseñador de gráficos creado especialmente para sacar el máximo partido a las posibilidades gráficas del 128. Ofrece gran variedad de comandos que hacen de él un programa sencillo de manejar a la vez que permite realizar unos diseños sorprendentes.

The Writer (Softek International): excelente procesador de textos elegido por la prestigiosa revista «Your Sinclair» como el mejor para el Spectrum. Compatibilidad con otros procesadores como Wordstar y Tasword.

Knight Time (Mastertronic): continuación del magnífico Spellbound. Una estupenda aventura gráfica.

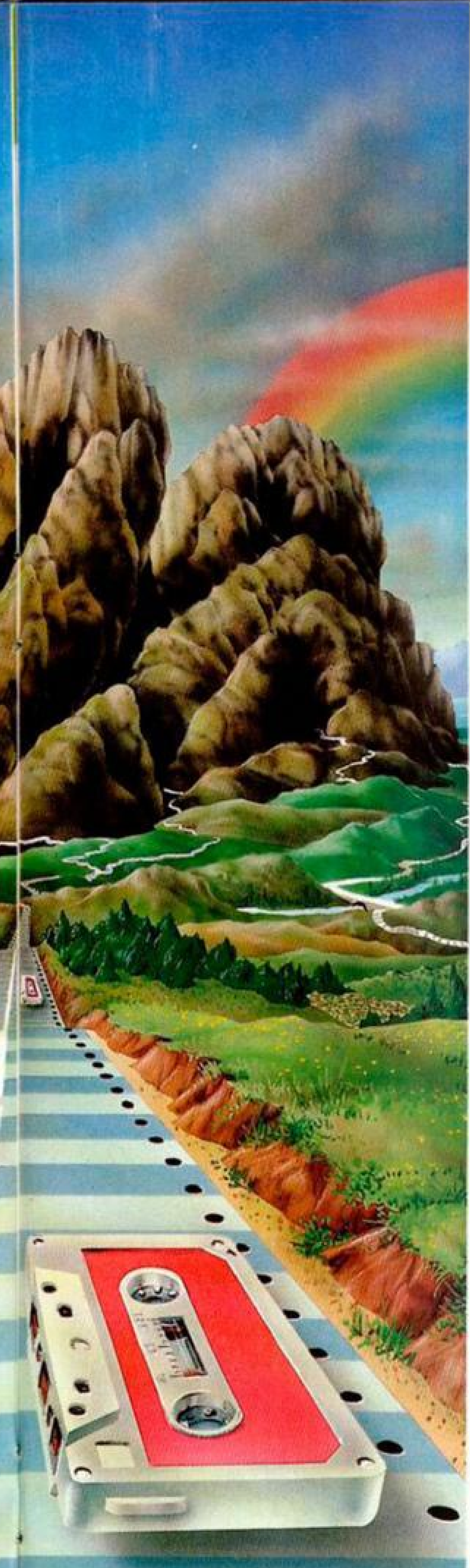
Wham¡ (Melbourne House): un nuevo y revolucionario programa de creación musical. Obtiene un gran rendimiento del sonido multicanal del 128, del mismo modo que permite la creación de diferentes melodías sin necesidad de tener conocimientos musicales.

Barry McGuigan World Championship Boxing: (Activision): un increíble programa de simulación deportiva, en este caso el boxeo, que permite crear nuestros propios púgiles, así como elegir su estilo de combate.

Lode Runner (Software Proyects): un divertido y original arcade que permite la creación de la escenas y componentes del juego.

Confrontation (Lothlorien): un wargame al estilo clásico pero que contiene 8 contiendas diferentes que van desde batallas históricas de la II Guerra Mundial, hasta conflictos actuales como los de Afganistán, Angola, etc...

Magnus Z80 Assembler / Monitor /



Analyser (Oasis Soft): este excepcional programa permite el control total de los desarrollos en código máquina. Un excelente Ensamblador.

Return to Eden (Level 9 Computing): aventura de texto que se desarrolla en más de 200 escenarios diferentes. Una emocionante aventura espacial.

Ball Blazer (Activisión): un simulador muy especial de una nave espacial. Una impresionante carrera galáctica.

Tasword 128 (Tasman Soft): una versión ampliamente mejorada del ya famoso Tasword para el Spectrum 48 K. Un nuevo avance para el procesamiento de textos.

I.C.U.P.S. (Thor Computer): simulación de una auténtica batalla espacial, que lleva al jugador a una dura prueba de habilidad y reflejos.

Sweevo's Whirled (Gargoyle): primero fue Sweevo's World, y ahora Gargoyle nos presenta la aventura completa de este torpe y feo robot en su odisea espacial particular.

Megamix Technician Ted (Hewson): una versión especialmente ampliada del exitoso arcade para Spectrum 48. Ahora con más de 100 pantallas y una más perfecta realización.

Fairlight 2 (Softek Int): The Edge presenta «La prueba de la oscuridad». Por fin, una segunda parte del Fairlight, programa que fue elegido uno de los diez mejores, incluido en la lista que contenía a los diez mejores juegos jamás realizados para el Spectrum.

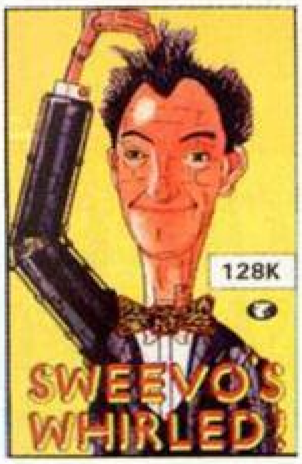
Dan's back in (Mirrorsoft): nuevos peligros amenazan al mundo. Dan volverá a demostrar sus habilidades en una nueva y sorprendente vidioaventura.

Y esta es la lista de los programas que hasta la fecha han sido confeccionados expresamente para el 128 y que suponen una total novedad. Sin embargo, existen muchos otros que sin llegar a ser completamente nuevos, suponen la conversión de ciertos éxitos anteriores.

Estas versiones pretenden potenciar al máximo las posibilidades gráficas y sonoras que el nuevo ordenador puede ofrecer, y entre estos programas se encuentran los siguientes títulos:

Gladiator
The Never Ending Story
Bored of the Rings
I, of the Mask
The Rocky Horror Show
Rasputín
Robin of the Wood
Daley Thompson's
3 Weeks in Paradise
Desert Rats
Winter Games

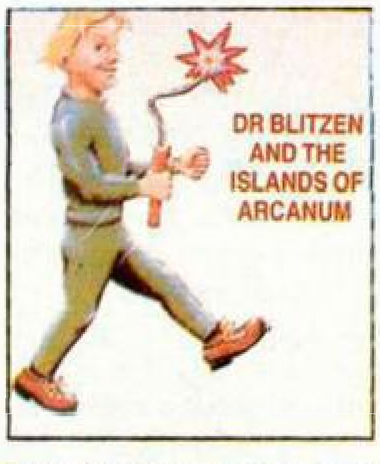
Y ya para acabar, simplemente comentar que la práctica totalidad de los programas del S 48 son compatibles con el 128, por lo que podíamos afirmar que este nuevo modelo es el que posee en la actualidad una mayor cantidad de software disponible.

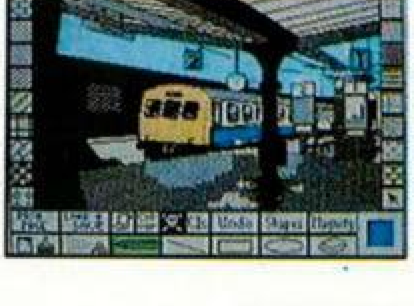












CHEQUEO DE LA MEMORIA EN EL SPECTRUM (Y II)

Miguel SEPULVEDA y Adolfo PEREZ



La semana pasada iniciábamos la descripción de esta interesante rutina que nos permite chequear el estado de la memoria de nuestro Spectrum y nos informa de los errores en caso de que los hubiera. Ahora continuamos con la explicación del funcionamiento, el cual podéis seguir, paso a paso, consultando el listado de su desensamble.

El programa consta de tres partes en total. La primera es el bloque de Basic que nos pregunta la dirección de memoria donde queremos ubicar la rutina principal (MEMORY). La segunda parte es la rutina principal propiamente dicha, es la encargada de chequear la memoria y, finalmente, la pantalla de presentación que es generada con el programa que presentamos esta semana (programa 2). Estos tres bloques programa 1 , la rutina MEMORY y pantalla, *deben ubicarse* por este orden en la cinta de casette, y que cada uno cargue la siguiente.

La rutina de chequeo de la memoria tiene un funcionamiento muy peculiar y que se autoubica con objeto de poder comprobar incluso las direcciones de memoria donde reside ella misma. En las dos pasadas de chequeo que realiza, su funcionamiento es distinto y podéis comprobarlo con ayuda del listado adjunto.

Primera entrada

Lo primero que se hace es inicializar los registros: El registro BC se inicializa a cero. Su contenido nos va a servir como indicador de las pastillas encontradas en fallo. El registro HL lo inicializaremos con la dirección a la cual desplazaremos el programa después de su primera ejecución. En el DE introducimos el contenido de la variable SEED y que será la última dirección de memoria a probar. Estos tres valores se guardarán en los registros alternativos. También se inicializa el registro A con el primer valor de prueba (que será A5). Con este valor se llama a la rutina que comprueba la memoria (entrada RELCO1). Esta rutina, calcula en primer lugar la memoria que hay delante del programa, que dependerá de la posición de carga. A continuación, rellena el trozo *a probar* con el valor contenido en el registro A y comienza a leer el trozo de memoria rellenado. Se va comprobando byte a byte el contenido de A con el contenido de la dirección *apuntada por el registro* HL. Si en cualquier momento ambos valores no coinciden, se hará una llamada a una rutina de error que nos irá apuntando las direcciones de error a partir de la posición 23296 (es la dirección que antes habíamos llamado X). El bucle de comparación estará controlado por los registros BC y HL que iremos decrementando e incrementando respectivamente. El registro simple C cuyo contenido serán números decrecientes desde 255 hasta cero, lo utilizaremos para ir sacando por el puerto 254 esos valores, y que nos producirán unos raros efectos de BORDER'S y sonidos. Cuando el registro BC alcance el valor cero, la comparación habrá terminado. Si el valor probado ha sido el hexadecimal A5, calcularemos el valor complemen-

PROGRAMA 2 - GENERADOR DE PANTALLAS

```
   1 BORDER 0: PAPER 5: INK 7: C
LS : GO SUB 400: GO TO 50
  10 PLOT x,y: DRAW w,0: DRAW 0,
-z: DRAW -w,0: DRAW 0,z: RETURN
  15 FOR n=1 TO a: READ x: GO SU
B 10: NEXT n: RETURN
  20 DATA 15,31,47,63,79,95,111,
127
  21 DATA 159,175,191,207
  22 DATA 119,135,151
  23 DATA 55,71,231
  25 DATA 15,168,33,25, 151,168,
73,9, 79,152,17,49, 175,152,17,4
9, 207,152,17,41
  40 PRINT PAPER 0;AT l,c;CHR$ a
;AT l+1,c;CHR$ b;AT l+2,c;CHR$ 1
45: RETURN
  50 LET a$="
"
  60 FOR n=1 TO 19: PRINT PAPER
4;AT n,1;a$: NEXT n
  80 LET x=7: LET y=168: LET w=2
41: LET z=153: GO SUB 10: PRINT
AT 0,3;"▬";AT 0,7;"Ā";AT 0,9;"Ā"
;AT 0,29;"A"
 100 RESTORE 20: LET y=48: LET w
=9: LET z=25: LET a=8: GO SUB 15
 110 LET y=56: LET a=4: GO SUB 1
5: RESTORE 21: LET y=88: GO SUB
15
 120 LET y=120: LET a=3: GO SUB
15: RESTORE 22: LET y=152: GO SU
B 15
 130 LET x=23: LET y=120: GO SUB
10
 140 LET y=168: LET w=9: LET z=w
: GO SUB 15
 150 FOR n=1 TO 5: READ x: READ
y: READ w: READ z: GO SUB 10: NE
XT n
 160 LET l=16: LET a=154: LET b=
146
 170 FOR c=16 TO 2 STEP -2: GO S
UB 40
 174 IF c>8 THEN LET b=b+1
 176 IF c<12 THEN LET a=a+1
 178 NEXT c
 180 LET l=7: LET c=3: GO SUB 40
 190 LET l=15: LET a=a+1
 192 FOR c=26 TO 20 STEP -2: GO
SUB 40: LET a=a+1: NEXT c
 194 LET l=11: LET c=26: GO SUB
40: LET a=155: LET b=b+1
 196 FOR c=24 TO 20 STEP -2: GO
SUB 40: LET a=a+1: NEXT c
 200 LET l=7
 202 FOR c=19 TO 15 STEP -2: GO
```



tario, lo meteremos en A y repetiremos el proceso. La operación será idéntica para los valores hexadecimales FF y su complementario (cero). Cuando el valor cero haya sido ya probado y, por tanto, la memoria anterior al programa ha sido puesta a ceros, guardamos la última dirección de error encontrada en el registro DE, en HL guardaremos la dirección de comienzo del programa y en BC su longitud. Con estos datos haremos LDIR con lo cual desplazaremos el programa desde la posición apuntada por HL hasta la dirección contenida en DE.

Luego, haremos un JP a la nueva dirección de arranque con lo que el programa empezará una nueva ejecución.

Segunda entrada

Esta segunda ejecución tendrá una diferencia fundamental con la primera.

Cuando probamos con el valor cero, la posición 16384 la pusimos a cero, de manera que cuando comience esta segunda ejecución y pregunte por el contenido de esta posición, el programa se irá por la segunda rama, en donde se repite el mismo proceso que se hizo por la primera probando con los valores hexadecimales A5, 5A, FF y Ø. A continuación se calcula la dirección de memoria donde cargaremos la pantalla (4ØØØH), la longitud en bytes a cargar (18ØØH) y se prepara en el registro A el indicativo de carga de datos sin cabecera, datos necesarios para llamar a la rutina del sistema operativo que se encuentra en la posición 556H. En el ángulo superior izquierdo de la pantalla correspondiente a la posición 58ØØH, nos aparecerá un cuadrado intermitente en blanco que nos indicará que hemos de poner el cassette en marcha para cargar la pantalla conteniendo los dibujos de las pastillas y que previamente grabaremos como se explica a continuación.

```
SUB 40: LET a=a+1: NEXT c
 204 LET l=3: LET c=19: GO SUB 4
0: LET a=154: LET b=b+1
 206 LET c=15: GO SUB 40: LET c=
c+2: LET b=b+1: GO SUB 40
 210 LET a$="
 220 PRINT PAPER 0;AT 1,2;" TU "
;AT 2,2;"UHF ";AT 3,2;a$
 230 PRINT PAPER 0;AT 1,7;"E";AT
 1,9;"M";AT 1,19;"   SLOT   ";AT
 1,29;"A": PLOT 160,168: DRAW 0,-
7: PLOT 161,168: DRAW 0,-7
 240 LET a$="  UILCA1       ": LET
b$="  ZI8C02       "
 250 FOR l=3 TO 8: LET a=(l-2)*2
-1
 254 PRINT PAPER 0;AT l,10;a$(a
TO a+1);AT l,22;b$(a TO a+1)
 256 NEXT l
 260 LET a$="  RIOCM5  "
 262 FOR l=3 TO 7: LET a=(l-2)*2
-1
 270 PRINT PAPER 0;AT l,26;a$(a
TO a+1): NEXT l
 300 PRINT #1;AT 0,0;"Pulsa tecl
a para salvar SCREEN$ "
 310 PAUSE 0: PRINT #1;AT 0,0;"
    PULSE [illegible] PARA SALIR      "
 320 RANDOMIZE USR 23300: REM Sa
lva SCREEN$
 330 STOP
 400 RESTORE 500: LET b=0
 410 FOR n=USR "a" TO USR "u"+7:
 READ a: POKE n,a: LET b=b+a: NE
XT n
 420 FOR n=23300 TO 23311: READ
a: POKE n,a: LET b=b+a: NEXT n
 430 IF b<>8131 THEN PRINT FLASH
 1;"ERROR EN DATAS": STOP
 440 RETURN
 500 DATA 0,0,0,0,0,126,126,255,
66,60,0,66,126,66,0,0,36,74,74,6
0,0,0,36,66,112
 501 DATA 72,68,66,0,0,36,66,44,
82,82,44,0,0,36,66,60,82,82,36,0
,0,36,66,2,126
 502 DATA 34,16,0,0,36,66,50,74,
70,34,0,0,36,66,44,82,66,36,0,0,
36,66,10,126,42
 503 DATA 24,0,0,36,66,60,24,0,0
,0,0,0,0,60,24,0,60,66,66,60,0,6
0,24,0,2
 504 DATA 126,34,16,0,60,24,0,50
,74,70,34,0,60,24,0,44,82,66,36,
0,60,24,0,10,126
 505 DATA 42,24,0,60,24,0,76,82,
82,116,0,60,24,0,36,74,74,60,0,6
0,24,0,112,72,68
 506 DATA 66,0,60,24,0,44,82,82,
44,0,60,24,0,60,82,82,36,0
 510 DATA 221,33,0,64,17,0,27,62
,255,195,194,4
```

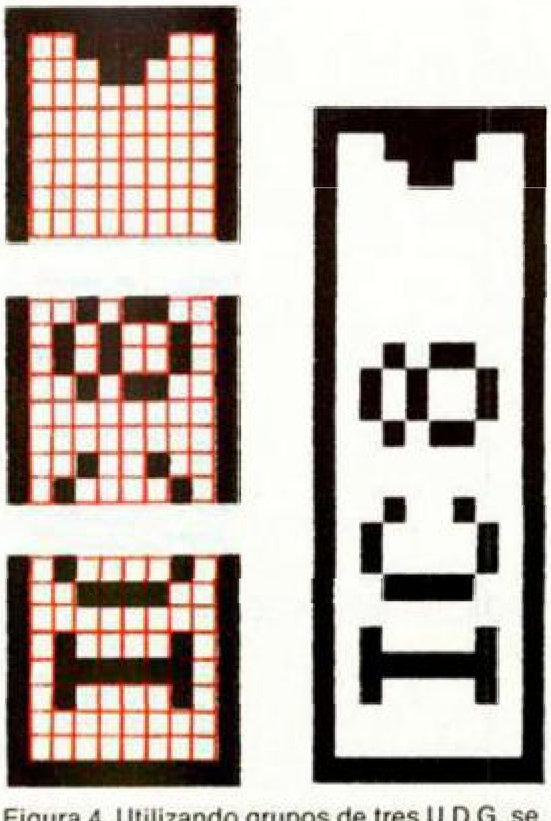


Figura 4. Utilizando grupos de tres U.D.G. se define la forma de los integrados que presentan ese aspecto en la pantalla.

El programa generador de la pantalla

El programa encargado de dibujarnos la pantalla (programa 2) con los componentes electrónicos del micro realiza un esquema en el que podemos distinguir todas las pastillas de circuitos integrados con sus correspondientes rotulaciones y los conectores.

La pantalla aparecerá en la TV tal como se muestra en la figura 5. El progra-

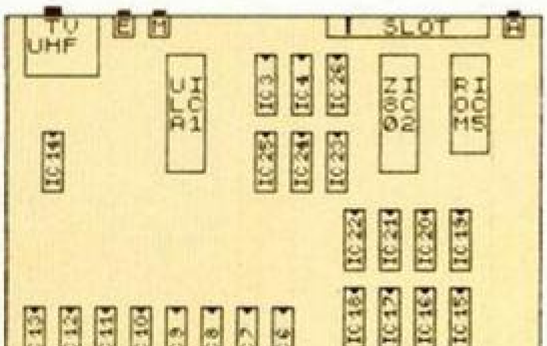


Figura 5.

ma aparentemente parece un poco complejo por la cantidad de dibujos que realiza, en distintas fases. Primeramente se crean unos gráficos cuya composición dará lugar a las diferentes pastillas de circuitos integrados. A continuación, se rellenan las pastillas con sus correspondientes rótulos. El programa «generador de pantalla» crea los gráficos y contiene una rutina C.M. que permite grabar en cinta la pantalla sin cabecera que

se ubica a partir de la dirección 233ØØ y se encuentra en los DATAS de la línea 51Ø. Si al final de todos los DATAS su suma no es 8131 nos dará un mensaje de error deteniendo la ejecución del programa en la línea 43Ø.

Una vez generada la pantalla, en la línea 23 del monitor TV aparecerá el mensaje:

PULSE CUALQUIER TECLA PARA SALVAR SCREEN$.

En este momento debemos poner el cassette en modo grabación para poder salvar la pantalla. El programa entra en un PAUSE hasta que demos a cualquier tecla, hecho lo cual, se llama a la rutina que está cargada en la posición 233ØØ para salvar la pantalla y el programa termina su ejecución con un Stop.

El resultado final

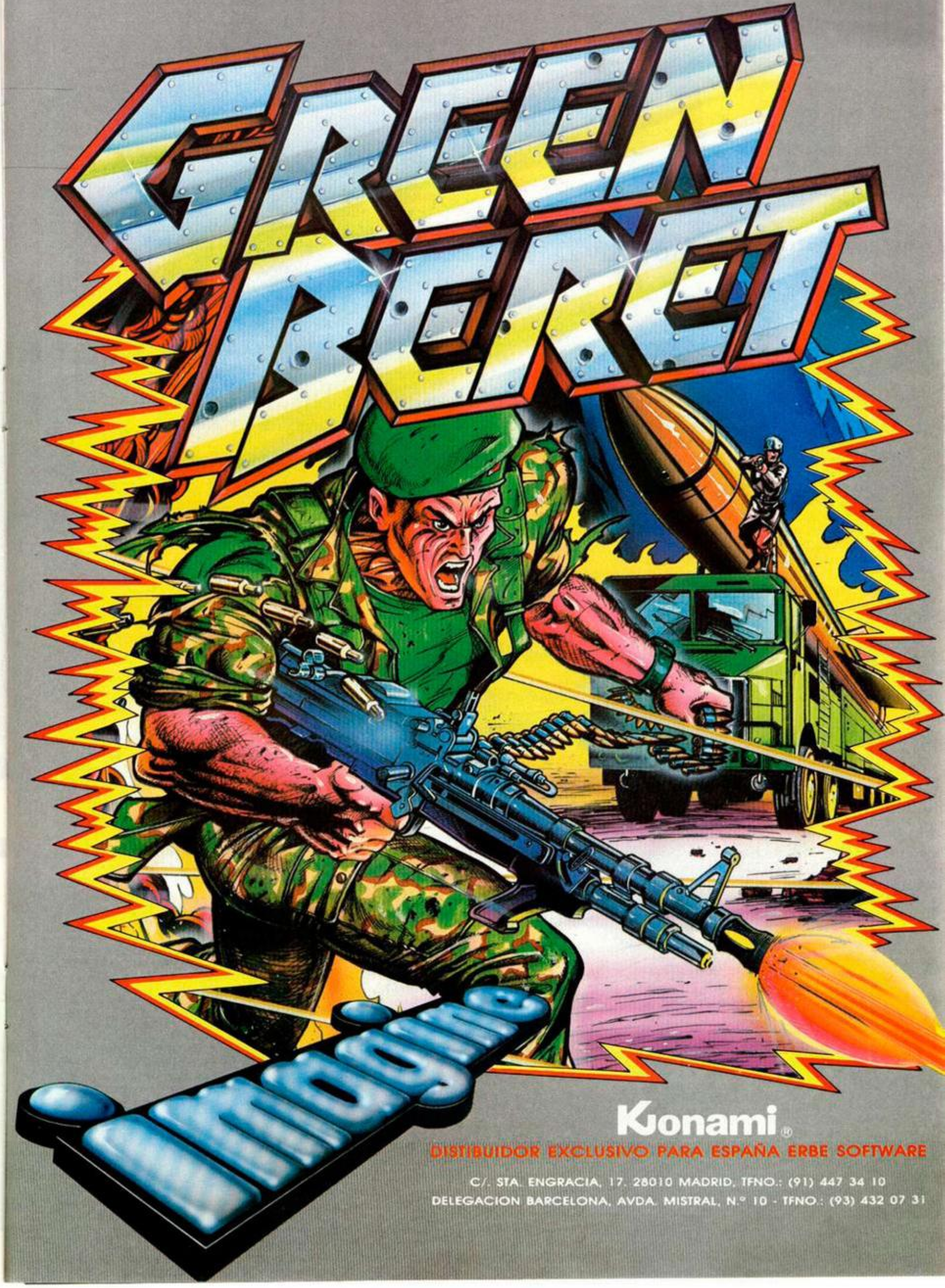
Cuando la pantalla haya sido cargada, y si durante la prueba no se encontró ningún fallo, se permitirán interrupciones y se entrará en un bucle de lectura de la tecla BREAK. Cuando esto se produzca, el programa saltará la posición cero reseteando el sistema.

Si se ha producido algún error, se calcula en HL la dirección de las pastillas a las que se debe el error y se las pone en modo intermitente sobre el dibujo de la pantalla cargada anteriormente. Luego, como en el caso anterior, se permiten interrupciones y se entra en el bucle de espera de la tecla BREAK.

Esperamos que os resulte de utilidad y evite más de un susto injustificado.

DESENSAMBLE DE LA RUTINA «MEMORY»

```
10 ;
20        ORG  #0000
30 ;
40 MEMORY EQU  $
50        CALL #52
60        DEC  SP
70        DEC  SP
80        EX   (SP),IX
90        DEC  IX
100       DEC  IX
110       DEC  IX
120       DEC  IX
130       LD   SP,IX ;Dir. del STACK= MEMORY-1
140       INC  IX ;Dir. origen del programa
150       LD   A,(#4000)
160       AND  A ;?Se ha desplazado el programa?
170       JR   Z,MEM2 ;Si
180       DI   ;** PRIMERA ENTRADA **
190       EXX
200       LD   BC,0 ;Pastillas que fallan
210       LD   DE,(23670) ;Ultima dir. de memoria
220       LD   HL,23296 ;Dir. donde se desplaza
230       EXX
240       LD   A,#A5 ;Comprueba ruidos entre BITS
250       CALL #52
260       JR   RELCO1
270       LD   A,#FF ;Comprueba BITS
280       CALL #52
290       JR   RELCO1
300       EXX
310       PUSH HL
320       EXX
330       POP  HL
340       LD   BC,20
350       ADD  HL,BC
360       PUSH HL
370       LD   BC,NUMMEM ;Long. del programa
380       PUSH IX
390       POP  DE
400       EX   DE,HL
410       LDIR ;Desplaza el programa
420       POP  HL
430       JP   (HL) ;Ejecuta el programa desplazado
440 ;
450 MEM2  EQU  $ ;** SEGUNDA PARTE **
460       LD   A,#A5 ;Comprueba ruidos entre BITS
470       CALL #52
480       JR   RELCO2
490       LD   A,#FF ;Comprueba BITS
500       CALL #52
510       JR   RELCO2
520       EXX
530       PUSH BC
540       EXX
550       LD   A,#87
560       LD   (#5800),A ;Pone en FLASH 1. caracter
570       PUSH IX
580       LD   IX,#4000
590       LD   DE,#1800
600       LD   A,#FF
610       SCF
620       CALL #556 ;Carga la pantalla
630       POP  IX
640       POP  BC
650       LD   A,B
660       OR   C ;?Hay fallo?
670       JR   Z,MEM20 ;No
680       PUSH IX
690       POP  DE
700 *H PONE EN FLASH LAS PASTILLAS QUE HAN FALLADO.
710 MEM12 EQU  $
720       LD   HL,TABATT
730       ADD  HL,DE
740       LD   A,B
750       LD   B,16
760 ;
770 MEM14 EQU  $
780       SRL  C
790       RRA
800       JR   NC,MEM16
810       PUSH HL
820       LD   E,(HL)
830       INC  HL
840       LD   D,(HL)
850       LD   HL,32
860       EX   DE,HL
870       PUSH AF
880       PUSH BC
890       LD   B,3
900 ;
910 MEM15 EQU  $
920       LD   A,#80
930       OR   (HL)
940       LD   (HL),A ;Pone pastilla en FLASH
950       ADD  HL,DE
960       DJNZ MEM15
970       POP  BC
980       POP  AF
990       POP  HL
1000 ;
1010 MEM16 EQU $
1020      INC  HL
1030      INC  HL
1040      DJNZ MEM14
1050 *H SALE CON UN BREAK RESETEANDO EL SISTEMA.
1060 MEM20 EQU $
1070      EI
1080      CALL #1F54
1090      JP   NC,0
1100      JR   MEM20
1110 *H RUTINA QUE COMPRUEBA LA MEMORIA
1120 ;
1130 RELCO1 EQU $
1140      DEC  SP
1150      DEC  SP
1160      LD   DE,#4014
1170      PUSH IX
1180      POP  HL
1190      AND  A ;Calcula memoria delante del
1200      SBC  HL,DE ;programa para probar
1210      LD   B,H
1220      LD   C,L
1230      LD   E,0
1240      JR   RELCOM
1250 ;
1260 RELCO2 EQU $
1270      DEC  SP
1280      DEC  SP
1290      PUSH IX
1300      POP  HL
1310      LD   DE,NUMMEM
1320      ADD  HL,DE
1330      PUSH HL
1340      EXX
1350      PUSH DE
1360      EXX
1370      POP  HL
1380      POP  DE
1390      AND  A ;Calcula memoria posterior del
1400      SBC  HL,DE ;programa para probar
1410      LD   C,L
1420      LD   B,H
1430 ;
1440 RELCOM EQU $
1450      EX   (SP),HL
1460      INC  HL
1470      INC  HL
1480      EX   (SP),HL ;Prepara retorno
1490      PUSH BC
1500      PUSH DE
1510 ;
1520 RC0   EQU  $
1530      PUSH DE
1540      LD   H,D
1550      LD   L,E
1560      INC  DE
1570      LD   (HL),A
1580      PUSH BC
1590      DEC  BC
1600      LDIR ;Rellena memoria
1610      POP  BC
1620      POP  HL
1630      LD   E,A ;Guarda caracter a comprobar
1640 ;
1650 RC2   EQU  $
1660      LD   A,E
1670      CP   (HL) ;Comprueba memoria
1680      JR   NZ,ERROR
1690 ;
1700 RC4   EQU  $
1710      INC  HL
1720      DEC  BC
1730      LD   A,C
1740      OUT  (254),A ;Cambia BORDER y SONIDO
1750      OR   B
1760      JR   NZ,RC2
1770      LD   A,E
1780      AND  A ;?Ha acabado?
1790      RET  Z ;Si
1800      CP   #5A
1810      RET  Z ;Si
1820      CPL  ;No. Hace otra pasada
1830      POP  DE ;complementando el dato
1840      POP  BC
1850      JR   RC0
1860 ;
1870 ERROR EQU  $
1880      PUSH HL
1890      LD   A,(HL)
1900      XOR  E
1910      BIT  7,H ;?Esta en los 32 K. ultimos?
1920      EXX
1930      JR   Z,ER0 ;No
1940      OR   C
1950      LD   C,A ;Pone bit de pastilla mala
1960      JR   ER2
1970 ;
1980 ER0   EQU  $
1990      OR   B
2000      LD   B,A ;Pone bit de pastilla mala
2010 ;
2020 ER2   EQU  $
2030      POP  HL
2040      EXX  ;Guarda ultima dir. que falla
2050      JR   RC4;
2060 *H DIR. DE LOS ATRIBUTOS DE LAS PASTILLAS
2070 TABATT EQU $-MEMORY
2080      DEFW #5A10 ;BIT 0
2090      DEFW #5A0E ;BIT 1
2100      DEFW #5A0C ;BIT 2
2110      DEFW #5A0A ;BIT 3
2120      DEFW #5A08 ;BIT 4
2130      DEFW #5A06 ;BIT 5
2140      DEFW #5A04 ;BIT 6
2150      DEFW #5A02 ;BIT 7
2160      DEFW #59FA ;BIT 0
2170      DEFW #59F8 ;BIT 1
2180      DEFW #59F6 ;BIT 2
2190      DEFW #59F4 ;BIT 3
2200      DEFW #597A ;BIT 4
2210      DEFW #5978 ;BIT 5
2220      DEFW #5976 ;BIT 6
2230      DEFW #5974 ;BIT 7
2240 ;
2250 NUMMEM EQU $-MEMORY ;** Longitud del programa **
```


GREEN BERET
imagine
Konami®
DISTIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE
C/. STA. ENGRACIA, 17. 28010 MADRID. TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TFNO.: (93) 432 07 31

MICRO *Manía*

Sólo para adictos

SIR FRED

El indudable éxito que ha conseguido este sensacional programa no queda solamente reflejado por la gran cantidad de votos que recibimos para incluirlo en nuestra lista de Microhits, sino también en la avalancha de cartas que nos llegan contándonos diferentes tipos de trucos y estrategias a seguir para conseguir rescatar a la bella princesa.

Por eso, hemos seleccionado algunas de ellas que nos han parecido más interesantes y con las que seguro que os resultará mucho más cómodo y divertido.

Javier Aranaz y Urko Sánchez, nos comentan que si nos suicidamos 10 veces seguidas (lo que se consigue apretando BREAK y posteriormente CAPSHIFT y 1), lograremos que aparezcan a lo largo de la partida una considerable cantidad de pollos, por lo que la energía estará asegurada. Entonces tendremos que ir en busca de las cerillas que se encuentran defendidas por un espadachín, a la izquierda de la gran escalera. Una vez que las hayamos conseguido, tenemos que subir por el órgano y en la pantalla de arriba se explota la bomba. Subimos por las escaleras hasta arriba del todo y pulsamos la palanca que hace bajar parte del techo. Saltando por el hueco de la escalera podremos subir por una cuerda, y una vez que hayamos dejado atrás a la nube, pasaremos las almenas por la izquierda, con lo que no tendremos más que matar al guarda que vigila a la princesa y rescatarla.

Otro truco que también nos han enviado muchos lectores consiste en que, una vez que nos encontremos en la pantalla de la nube que tiene una palanca, en lugar de pulsarla, lo que podemos hacer es tomar mucha carrerilla y saltar hacia la izquierda, con lo que apareceremos en la pantalla en la que un espadachín se encuentra defediendo un puente levadizo.

Sin embargo, Dionisio Bustillo ha ido mucho más allá, y desde Melilla nos ha enviado un camino que, basándose en este detalle, afirma ser la manera más rápida de rescatar a la princesa.

Esta consiste en, una vez que nos encontremos frente a dicho espadachín, bajar a la puerta rápidamente, dar un paso hacia el guardia y luego un salto largo, es decir, agachándonos previamente. Así habremos conseguido encararnos en la cadena. Subiendo por ella daremos un nuevo salto llegando al final y apareceremos en una nueva pantalla en la que se encuentran una rata y un arquero. Entonces avanzaremos hacia la izquierda y podremos ver a la princesa en lo alto de un andamio. Para llegar hasta ella, bastará con apretar una palanca y saltar la plataforma que nos llevará a su encuentro.



Por último, os ofrecemos por gentileza de Eduardo Méndez, el siempre codiciado POKE de vidas infinitas, aunque tenemos que advertir que sólo sirve para las copias que no sean en carga rápida.

POKE 46647, 201

¡Suerte, caballero!

LIBROS



BANCO DE DATOS

S. Rodgers y M. Milton

Ed. Generales Anaya, 45 págs.

En este volumen de la serie «Aprende a programar» nos encontramos con una aventura conversacional que se desarrolla en un mundo submarino. El jugador utiliza verbos y nombres para indicar qué es lo que desea llevar a cabo y el ordenador busca estas acciones en una Base de datos para interpretarlas correctamente. De ahí el nombre del libro.

La primera parte desarrolla el programa principal y todos los datos necesarios, y la segunda explica cómo los verbos que utiliza el jugador llevan al ordenador a ejecutar las diversas acciones.

Al principio hay una descripción del plano utilizado en esta aventura, dividido en 24 habitaciones distintas. En ellas puede apreciarse el mundo submarino donde se va a desarrollar la acción, los distintos peligros y tareas con las que deberemos enfrentarnos, así como una lista con las palabras y verbos que podemos utilizar.

Más tarde pasa a describir la introducción de los DATAs que van a servir de parámetros para el juego. Lo más interesante es que, además, explica cómo modificarlos para crear nuestros propios juegos.

El programa principal, que es el que gestiona toda la información proporcionada por los DATAs, está descrito paso a paso, con indicación de las acciones que se llevan a cabo en cada momento y las variables utilizadas.

Esto es muy útil a la hora de desarrollar nuestros propios programas.

Luego hace una aproximación a las técnicas utilizadas en los programas conversacionales para el tratamiento e interpretación de los verbos y cadenas de caracteres, finalizando con los listados de los programas para Spectrum y Commodore.

El libro no está pensado para jugar, ya que en las explicaciones sobre el juego acaba descubriendo los enigmas del mismo, sino para aprender a crear juegos de aventuras conversacionales y divertirnos con nuestros amigos inventando nuestras propias aventuras.

Cómo se guarda un número en un listado

LA BIBLIA DEL HACKER (IV)

En el capítulo anterior comentábamos la posibilidad de modificar un listado Basic de forma que confunda a cualquiera que trate de inspeccionarlo a la vez que su funcionamiento es perfectamente correcto. Una de estas posibilidades es alterar los valores ASCCI de las cifras numéricas.

Imaginemos una línea de Basic tal como: 1Ø LET a = 1ØØ. El número cien se guarda en la memoria de dos formas distintas: primeramente los códigos ASCII del 1, y los dos Ø, luego el prefijo 14, que indica que los próximos 5 octetos son la representación del número en coma flotante, y a continuación los cinco octetos de esta representación.

En el ejemplo se guardaría de la siguiente forma: 49, 48, 48, 14, Ø, Ø, 1ØØ, Ø, Ø. La representación ASCII se utiliza a la hora de presentar el número en la pantalla, y los cinco octetos en coma flotante se usan a la hora de los cálculos que realiza el ordenador.

Si reflexionamos sobre esto nos daremos cuenta de que no hay nada que impida que en la pantalla se imprima un número y luego, a la hora de considerarlo en un cálculo, sea totalmente distinto. Bastaría con hacer un Poke en la dirección que contiene el 1ØØ, por ejemplo con 2ØØ, para que al ejecutar la línea de Basic con un RUN la variable «a» se actualice con el valor 2ØØ, y sin embargo, en el listado se ve un 1ØØ claramente. Esto, además, tiene la siguiente particularidad: si editamos la línea 1Ø y la volvemos a introducir en el listado con la tecla «Enter», la representación en coma flotante se ajusta automáticamente a los valores indicados por los códigos ASCII con lo que la línea ya no es lo que era. Esta protección se conoce con el nombre de «Literales ASCII retocadas.»

Obviamente existe el efecto contrario, es decir, que en vez de «pokear» en la representación del número en forma decimal a la hora de hacer la protección se modifique el literal ASCII.

De todo lo arriba expuesto se deduce que debemos buscar alguna forma de ver un listado sin que por ello se modifique sustancialmente.

El programa COPYLINE

En la revista número 3 se publicó un programa, COPYLINE, original de José María Reus, al que el autor de esta serie le ha hecho algunas modificaciones para que se adapte mejor a este caso concreto. Tecleamos el nuevo listado (programa 1), lo sálvamos en cinta y lo guárdamos muy bien pues lo vamos a tener que usar intensivamente. Un consejo: si tenemos un buen compilador hacemos lo propio con el programa y obtendremos unos resultados increíbles.

Con el presente programa se pueden ver cargadores de Basic sin tener que ubicarlos en la zona del Basic.

Para ello, en primer lugar se ha de modificar la cabecera del cargador Basic para convertirla en bytes y poder cargarlo en otra dirección, la manera de hacer esto es con el «Copyupi» publicado en los números 44 y 45.

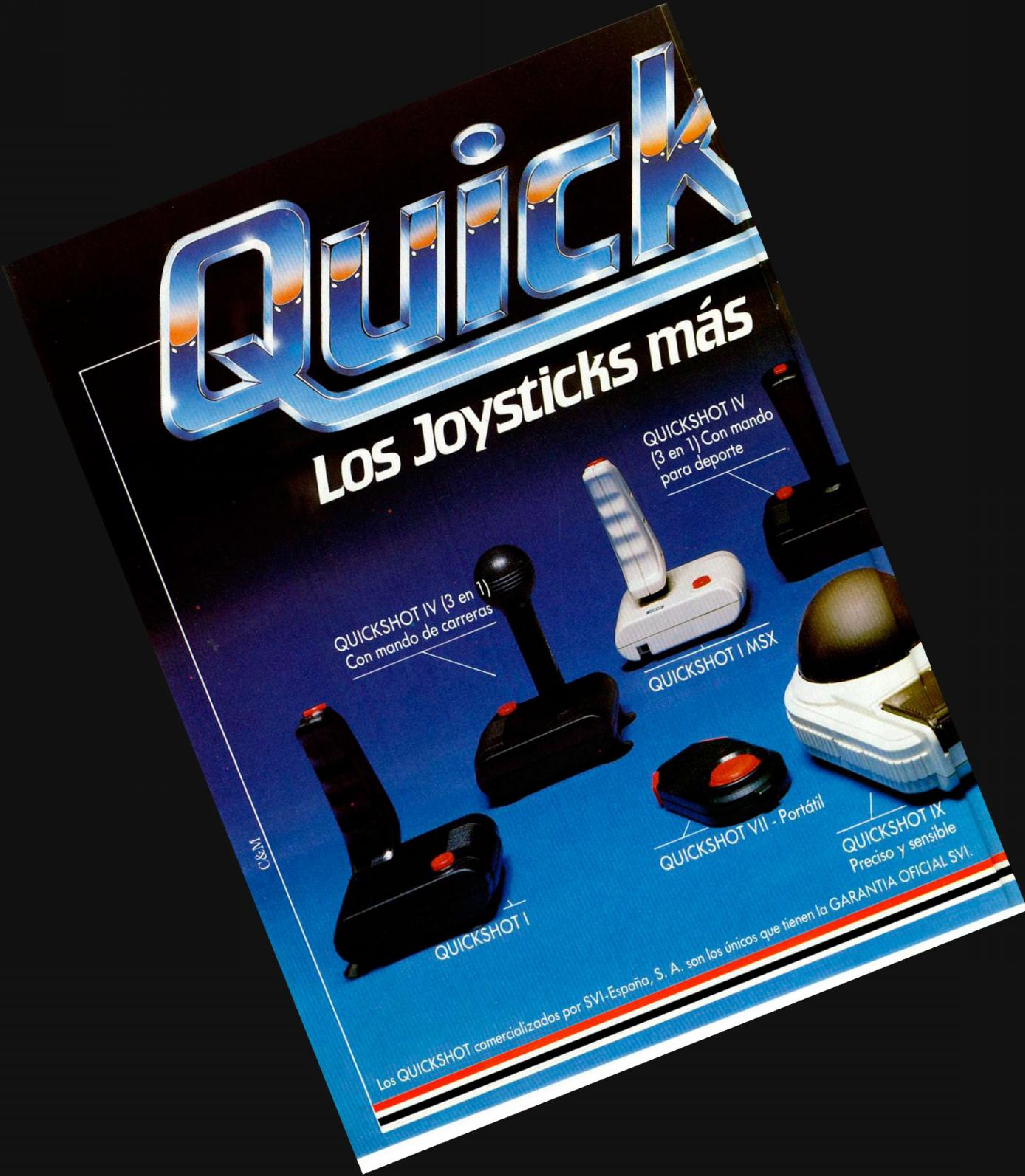
Cargamos el programa con la opción «LN» y luego, con la opción «CC» cambiamos los datos de la cabecera. El dato número 1 (tipo), pasará a bytes en lugar de programa y el dato número 4 (comienzo), pasará a ser cualquier posición de memoria que vayamos a tener libre, por ejemplo, la 3ØØØØ. Por supuesto, luego deberemos grabar en cinta el nuevo cargador modificado.

En este punto ya sólo queda cargar en memoria el Copyline, y haciendo un Break, cargar el programa modificado en la dirección 3ØØØØ, por ejemplo. Damos RUN al Copyline y respondemos a las preguntas que nos hace con Ø, para la primera línea del listado, 9999 para la última, y 3ØØØØ, la dirección donde hemos cargado el Basic, para la tercera pregunta. En el caso de que el cargador tuviera una línea de auto-run distinta a la Ø habría de darla como respuesta a la primera pregunta del Copyline.

El programa nos lista un Basic que esté ubicado en otra dirección aunque tenga cualquier protección de controles de color o cursor. El listado lo produce en 5 columnas, la primera indica la posición de memoria que se está explorando, en este caso esta posición no nos vale para nada ya que, recordemos, hemos ubicado el cargador en otro sitio. Las dos columnas siguientes nos informan del número de línea que se está explorando y la longitud en octetos de la misma.

Es en las dos últimas columnas en donde deberemos centrar nuestra atención: la antepenúltima indica el valor del byte dentro del programa y la última, la más importante, puede indicar varias cosas: o bien el TOKEN que se halle en el listado, o bien nada si el valor de octeto no es imprimible, o bien la representación VERDADERA de un argumento numérico que se halle dentro del listado. De esta forma no nos dejamos engañar por la protección de las literales ASCII retocadas.

```
10 BORDER 0: PAPER 0: INK 7: CLS
20 PRINT TAB 6; INVERSE 1;"LISTADO DE PROGRAMAS": PRINT '"DIT";"SENT";"LONG";"BYTE";"CODI o NUM": POKE 23689,PEEK 23689+1: PRINT OVER 1;"________________"
30 INPUT "Primera linea listada ";LINE a$
40 FOR I=1 TO LEN a$: IF a$(I)<"0" OR a$(I)>"9" THEN GO TO 30
50 NEXT I
60 IF a$="" THEN LET prs=10000: GO TO 90
70 IF VAL a$<10000 AND VAL a$>=0 THEN LET prs=VAL a$: GO TO 90
80 GO TO 30
90 INPUT "Ultima linea listada ";LINE a$
100 FOR I=1 TO LEN a$: IF a$(I)<"0" OR a$(I)>"9" THEN GO TO 90
110 NEXT I
120 IF a$="" THEN LET uls=10000: GO TO 150
130 IF VAL a$<10000 AND VAL a$>=0 THEN LET uls=VAL a$: GO TO 150
140 GO TO 90
150 INPUT "Direccion de comienzo? ";dir
160 LET nsc=PEEK (dir+1)+256*PEEK dir
170 LET lon=PEEK (dir+2)+256*PEEK (dir+3)
180 IF nsc<prs THEN LET dir=dir+lon+4: GO TO 160
190 IF nsc>uls THEN GO TO 530
200 LET pun=dir+4
210 PRINT dir;TAB 6;nsc;TAB 11;lon;TAB 17;PEEK dir
220 FOR i=1 TO 3
230 PRINT dir+i;TAB 17;PEEK (dir+i)
240 NEXT i
250 LET dir=dir+lon+4: LET ruca=0: LET runu=0
260 LET peek=PEEK pun
270 IF peek=13 AND pun=dir-1 THEN PRINT pun;TAB 17;peek: GO TO 160
280 IF peek=34 AND ruca=0 THEN LET ruca=1: GO TO 320
290 IF ruca=1 THEN GO TO 360
300 IF peek=14 AND runu=1 THEN GO SUB 380: LET runu=0: GO TO 260
310 IF peek<58 AND peek>47 THEN LET runu=1
320 PRINT pun;TAB 17;peek;
330 IF peek<32 THEN PRINT
340 IF peek>31 THEN PRINT TAB 21;CHR$ peek
350 LET pun=pun+1: GO TO 260
360 IF peek=34 THEN LET ruca=0
370 GO TO 320
380 PRINT pun;TAB 17;PEEK pun: LET pun=pun+1
390 DIM a(5): FOR i=1 TO 5: LET a(i)=PEEK (pun+i-1)
400 NEXT i
410 IF a(1)=0 AND (a(2)=0 OR a(2)=255) AND a(5)=0 THEN LET num=(a(3)+256*a(4))*(a(2)=0)+(a(2)=255)*((a(4)-256)*256+a(3)): GO TO 490
420 LET num=0
430 FOR i=5 TO 2 STEP -1
440 LET num=(num+a(i))/256
450 NEXT i
460 IF a(2)<128 THEN LET s=1: LET num=num+1/2
470 IF a(2)>=128 THEN LET s=-1
480 LET num=s*num*2↑(a(1)-128)
490 PRINT pun;TAB 17;PEEK pun;TAB 21;num
500 FOR i=1 TO 4
510 PRINT pun+i;TAB 17;PEEK (pun+i)
520 NEXT i: LET pun=pun+5: RETURN
530 INPUT "QUIERE CONTINUAR S/N";A$: IF A$="S" OR A$="s" THEN GO TO 1
540 STOP
```



Quick
Los Joysticks más
QUICKSHOT IV
(3 en 1) Con mando
para deporte
QUICKSHOT IV (3 en 1)
Con mando de carreras
QUICKSHOT I MSX
QUICKSHOT I
QUICKSHOT VII - Portátil
QUICKSHOT IX
Preciso y sensible
Los QUICKSHOT comercializados por SVI-España, S. A. son los únicos que tienen la GARANTIA OFICIAL SVI.
C&M



kShot®
vendidos del mundo.
QUICKSHOT IV (3 en 1)
Con mando para combate
QUICKSHOT II MSX
Con autodisparo
QUICKSHOT II
Con autodisparo
QUICKSHOT VII MSX
Portátil
Importador exclusivo SVI-España.
SVI
SPECTRAVIDEO

CONSULTORIO

Cuestión de columnas

He confeccionado un programa con unos textos que se incluyen en el listado del mismo y he aprovechado la subrutina que publicaron ustedes en el número 22 para escribir con 64 columnas.

Ahora quiero comprarme una impresora y ésta es mi pregunta: ¿Tengo que cambiar alguna sentencia de la línea 9000 o de la rutina en código máquina?

Emilio PALOMARES - Madrid

☐ La rutina para trabajar en 64 columnas sólo es útil para escribir sobre la pantalla. Si va a utilizar una impresora, no necesita la rutina para nada, ya que las impresoras tienen su propio juego de caracteres y trabajan, normalmente, con 80 o más columnas.

La única modificación que tendrá que hacer en el programa es incluir las oportunas sentencias «LPRINT» que le realicen el volcado de datos por la impresora.

Videojuegos

Os escribo para que me digáis qué tipo de lenguaje se utiliza para programar en las máquinas que se ponen en las salas de juego.

Además, me gustaría que me dijerais si existen versiones para Spectrum o Amstrad de los siguientes juegos de máquinas recreativas:

— Casa Nichibuso: Magmax, Galivan y Terra Cresta

— Casa Sésamo: Vastar

— Pinball y «99. The last war» (desconozco las casas). Además de Blish Blash (juego de demolición con bombas y tiempo).

ELECTRON MALDITO - San Sebastián

☐ Normalmente, los juegos de las máquinas recreativas están almacenados en una EPROM y se programan en Assembler. No obstante, en este tema existe mucha variedad por lo que no es posible generalizar.

En cuanto a los juegos que nos indica, no tenemos noticias de que existan versiones para Spectrum o Amstrad. Al menos, no en España.

Cassettes

Me han dicho que es mejor utilizar, para el ordenador, un cassette mono que uno stereo. Quisiera saber si el usar uno stereo puede ser perjudicial o si se puede causar alguna avería en mi ordenador.

Iñigo UYALDEBERE - Vizcaya

☐ La razón de recomendar cassettes monoaurales para el Spectrum es por que son más fiables y dan menos problemas de carga y verificación. De ninguna manera, puede el cassette dañar al ordenador. A lo sumo, serán los nervios del usuario los que resulten dañados.

«Micro-fichas»

Os escribo para preguntaros si en la micro-ficha R-0, que es la que lleva el cargador hexadecimal, habéis cometido algún error en el listado Basic, pues lo he tecleado varias veces y revisado también varias veces y no me funciona. Ocurre lo siguiente: Después de haber tecleado todo el programa realizo «RUN», entonces me pregunta la dirección, yo le doy 60000 y «ENTER» y, como resultado, me da: «E Out of DATA, 1070:2»

Angel RAMIREZ - Valencia

☐ El programa de la microficha R-0 funciona perfectamente, lo que ocurre es que no puede funcionar si no tiene las líneas «DATA» correspondientes a alguna de las micro-fichas R-n.

El procedimiento para utilizar este programa es teclearlo y guardarlo en cinta. Cada vez que se quiera cargar una rutina, se carga este programa y se le añaden las líneas «DATA» correspondientes a la rutina que se desee cargar. Evidentemente, si se intenta hacer funcionar el programa sin tener ninguna línea «DATA», lo lógico es obtener el informe: «Out of DATA».

La publicidad

El motivo de esta carta es manifestarles la opinión que me merece un aspecto que es el único descuidado de la revista: la PUBLICIDAD. Efectivamente, en sus páginas se observa mucha publicidad; es lógico puesto que si no, no podrían mantenerse. Pero el defecto es que en muchas de estas páginas de publicidad se ven, sistemáticamente, unas imperdonables erratas. Estos errores son de todo tipo: desde confundir vocales y crear programas inexistentes, a cambiar el nombre de un ordenador. (Núm. 69. Pág. 10: Amstrad CPC-472: un modelo que no tengo noticia de que los señores de esta compañía hayan sacado). Lo que quisiera pedirles es que revisasen un poco su publicidad.

José M. BELMONTE - Madrid

☐ Le agradecemos sinceramente su carta, aunque lamentamos no poder complacerle. Nuestros anunciantes nos entregan directamente los fotolitos de su publicidad, por lo que nosotros no podemos quitar ni poner siquiera un punto.

Respecto al anuncio concreto al que usted se refiere, no se trata de una errata. El Amstrad CPC-472 existe. Si desea información sobre esta máquina, puede dirigirse al anunciante (Micro-1) o a Indescomp.

Joystick Kempston

¿Se pueden imprimir puntos menores que un «PLOT»?

¿Me pueden decir cuáles son los números que hay que poner para que funcione con joystick Kempston la sentencia: «IF IN 31 = ...»?

Gonzalo FERNANDEZ - Madrid

☐ La menor unidad de información en un ordenador es un bit. El comando «PLOT» pone a «1» un bit del archivo de pantalla. Por tanto, es imposible imprimir puntos más pequeños que los que imprime el comando «PLOT».

El joystick Kempston no se lee por el port 31, sino por el 223. Los datos que devuelve son:

Arriba	: 8
Abajo	: 4
Derecha	: 1
Izquierda	: 2
Disparo	:16
En reposo	: 0

En nuestro artículo «Movimiento con teclado y joystick» se explicaban, con más profundidad, las distintas formas de leer un joystick.

Códigos de color en los listados

1. A veces interesa resaltar, en un listado, una sentencia REM sobre fondo blanco brillante; utilizando CAT (tecla 9) se consigue, pero se extiende al resto del listado. ¿Cómo puedo conseguir que no sea así?

2. ¿Cómo puedo conseguir ampliar un GDU más allá de las medidas de un carácter.

José CARRION - Córdoba

Si coloca un código de «Brillo especial» (Extendido «9») antes de la sentencia REM, puede anular el efecto colocando otro código de «Brillo normal» (Extendido «8») después de la misma.

Para hacer un GDU de más de 64 pixels, existen varios métodos; la mayoría de ellos en código máquina. La forma más sencilla de conseguirlo es utilizar dos o más caracteres definibles que deberán imprimirse juntos.

«Bismark»

Al hacer funcionar el programa «Bismark» del número 58, se me interrumpe con el mensaje: B Integer out of range, 3650:1. El listado está idéntico a como figura en la revista.

Jesús GONZALEZ - Avila

Suponiendo que la línea 3650 esté correctamente tecleada, las variables que pueden dar lugar al error son «BX» y «BY». Le recomendamos que revise las líneas 8000 a 8080 para comprobar si el error está en ellas, ya que es donde se asignan estas variables.

«Logo»

Estoy tecleando el «Logo» de los números 47 y 48. Al hacer el inicializador de variables, no me corre y no sé por qué. Lo pongo en marcha y, al empezar a cargar los «DATA», me sale: Variable not found, 30:5.

Jaime MARTI - Barcelona

Con frecuencia, nos escriben lectores diciendo que algún programa de nuestra revista no les funciona. En ocasiones es por un fallo de reproducción del mismo, otras veces el fallo es del lector al copiarlo. En cualquier caso, es imprescindible que nos indiquen el mensaje de error que da el programa y, sobre todo, los números que acompañan a dicho mensaje (línea y comando). Esto nos resulta necesario para localizar el posible error, si la culpa es nuestra, o para orientar al lector en su búsqueda, si la culpa es de él.

En este caso, parece ser que el programa no encuentra una variable en el quinto comando de la línea 30; vamos a ver a qué puede ser debido. En principio, esta línea sólo utiliza una variable: «a$». Sería demasiado extraño pensar que el lector se ha equivocado en el cuarto comando de la línea donde se lee esta variable; así que veamos otra posibilidad más factible. La función «VAL a$ (1)» deberá evaluar el primer carácter de la cadena leída en «a$», para lo cual este carácter ha de ser un número. Si fuera una letra, la función la tomaría por una variable, la buscaría y si no la encontrara, daría el informe. Como hemos hecho «RESTORE 350», estamos leyendo los «DATAs» que hay a partir de la línea 350. Cada uno de los literales de las líneas 350, 400 y 450 empieza por un número. Es muy posible que, en alguno de ellos, haya omitido ese número o haya tecleado una letra en su lugar. Revise las líneas 350, 400 y 450 porque es casi seguro que el fallo estará en alguna de ellas.

El depurado de errores en Basic no es difícil si se sigue una sistemática de trabajo como la que hemos seguido en este caso. Es muy frecuente cometer errores al hacer un programa o al teclearlo. Les pasa a los mejores programadores y nadie debe avergonzarse por ello. Además, el buscar los errores y solucionarlos es una tarea muy entretenida y que ayuda a aprender bastante.

«RTTY»

Quisiera saber si existe en el mercado algún interface-demodulador que, conectado a un equipo de radio, me permita transmitir en RTTY y el software que este dispositivo necesitaría.

Quisiera saber, además, si es posible transmitir programas por radio en BLU o AM o la manera de pasarlos por teléfono sin MODEM.

Jorge HERNANDEZ - Uruguay

Cada vez son más los lectores que nos escriben desde el otro lado del Atlántico (Argentina, Chile, Uruguay). Solemos contestarles personalmente porque es la forma más rápida de hacerles llegar las respuestas. No obstante, en este caso hemos considerado que la cuestión planteada podía tener bastante interés para otros lectores, por lo que hemos decidido incluirla en esta sección, además de enviarle, a este lector, la respuesta por correo.

Por el momento y que nosotros sepamos, no existe en el mercado ningún interface que permita hacer «RTTY» con el Spectrum. La razón tal vez haya que buscarla en el escaso número de usuarios potenciales que tendría este dispositivo. No obstante, tenemos noticias de que algunos usuarios han hecho experimentos en este sentido con resultados bastante satisfactorios aunque, eso sí, se han tenido que preparar tanto el software como el hardware necesario.

También hay quien conecta directamente la salida «MIC» a la entrada de una emisora y consigue transmitir programas. En este caso, los resultados dependen, fundamentalmente, del QRM que se tenga durante la transmisión. Si se modula en «BLU», es posible que se reduzca considerablemente el QRM aunque resultarían muy críticos los desplazamientos de frecuencia por lo que habrá que ser muy «fino» al ajustar el «clarificador».

Respecto a la transmisión telefónica, las líneas españolas tienen un nivel de «ruido» tan alto que, a veces, ni con MODEM es posible conseguir buenos resultados. No obstante, ignoramos la calidad de las líneas telefónicas uruguayas por lo que nos es imposible responder a esta pregunta.

Para los no versados en temas de radio-afición, incluimos un breve glosario de los términos empleados:

— RTTY: Sistema de transmisión radiofónica que consiste en modular la portadora con señales codificadas de ordenador, en lugar de hacerlo con morse o fonía.

— AM: Sistema de modulación que consiste en variar la amplitud de la portadora en función de la señal moduladora, obteniendo una frecuencia resultante compuesta por la portadora y dos bandas laterales correspondientes a la suma y diferencia de portadora y moduladora.

— BLU: También conocida como «SSB». Sistema de modulación en banda lateral única que consiste en eliminar la portadora y una de las bandas laterales tras la modulación en amplitud para conseguir que toda la potencia se emplee en enviar la señal moduladora. Se emplea, fundamentalmente, en transmisiones de onda corta a grandes distancias.

— QRM: Interferencias producidas durante la transmisión que empeoran la relación señal/ruido.

DE OCASION

● VENDO Spectrum 48K, en perfecto estado, con garantía, colección de revistas ZX, Todo-Spectrum y Microhobby, así como el libro de Código Máquina, manuales, cables, etc. Interesados llamar al tel. (93) 371 85 18 (a partir de las 3 tarde). Preguntar por Alfonso.

● ¿QUIERES comentar ideas, trucos, hablar de hardware? Si te interesa alguna de estas cosas no lo dudes, ponte en contacto con José Manuel Pérez. C/ Mayor, 71, 4.º 1.ª. Hospitalet de Llobregat. Barcelona. Tel. (93) 337 83 73.

● VENDO amplificador Optimus Art 30 W, 10 salidas exteriores. Entradas para micrófonos, plato, platina, etc. Su precio real es de 60.000 ptas. Lo vendo por 40.000 ptas. También lo cambio por Spectrum Plus más 5.000 ptas. Interesados llamar al tel. (981) 31 68 69. Preguntar por Carlos.

● VENDO ordenador Oric Atmos en perfecto estado con revistas, trucos personales, un joystick, cassette con todos los cables. Interesados escribir a la siguiente dirección: Luis García Arroyo. C/ Maragall, 38, 3.º, 3.ª. Ripollet. Barcelona.

● VENDO Spectrum 48K, por 25.000 ptas., y revistas nacionales e inglesas por 3.000 ptas. Interesados contactar con Carlos al tel. (91) 705 92 87.

● VENDO Spectrum 48K con reset y todos los conectores y manuales, cassette personal Computer Program Recorder, TV B/N 14" marca Sanyo, 2 libros uno de ellos de programación Basic. 1.º tomo de Run (enciclopedia práctica de Spectrum), revistas. Regalo 3 estuches y libro de Basic. Todo por 54.000 ptas. Interesados llamar al tel. (93) 311 10 47. Preguntar por Javier. Barcelona.

● VENDO Spectrum 48K Plus, en perfecto estado, incluyendo adaptador, toda de antena, manual en español e inglés, cinta de demostración. Precio: 35.000 ptas. Interesados contactar con Jordi al tel. (93) 338 58 97.

● VENDO Zx Spectrum 48K, en buen estado, con manuales en español y garantía. También vendo Zx Microdrive más interface 1 por 27.500 ptas. El ordenador y sus accesorios por 29.800 ptas. Interesados contactar con Raúl Díaz Medina. C/ Saliente, 59. Pozuelo de Alarcón. Madrid, tel. 715 30 59.

● VENDO ordenador Spectravideo SVI 318 con cassette SV 904, nuevo, por sólo 29.900 ptas. También vendo consola de vídeo-juegos Philips Videopac G.7000, nueva por sólo 14.900. Dirigirse a la siguiente dirección. Juan Carlos Castillo. C/ Maresma, 32. Tarrasa. Barcelona. Tel. (93) 783 67 37.

● VENDO Zx Spectrum 48K (20-4-85) más accesorios, interface Kempston, TV B/N de 22", 62 revistas de Microhobby, 1 libro de Código Máquina. Todo por sólo 90.000 ptas. Escribir a la siguiente dirección: Alberto González. Pinzón, 13, 3.º B. Madrid 28025.

● COMPRO Zx-81 sin ampliar. Revistas nacionales y extranjeras de informática en general. Interesados en la compra escribir a Andrés Blanco Baulo. Apdo. de correos 2168. Sabadell. Barcelona.

● VENDO Zx Spectrum 48K en perfecto estado, teclado DK'Tronics, Inferface 1, interface para joystick Kempston. Incluyo también los 30 primeros números de la revista Microhobby. Todo por 48.000 ptas. Interesados escribir a Benito Garrido Díaz. C/ Francisco Quero, 20. Porcuna. Jaén. Tel. (953) 54 43 30 (3 a 8 tarde).

● MULTIFACE-ONE, importamos periféricos directamente desde Inglaterra. Vendemos multiface-One con pulsador reset e interface de vídeo. Con instrucciones traducidas al castellano. Por sólo 10.500 ptas. Gastos de envió incluidos. Pedidos contra reembolso. Computers Log. Apartado 1212. Huelva. Garantía por 6 meses.

● NECESITO las instrucciones (fotocopias) de juego Broad Street. Las cambio por otras de otros programas, escribir a Alfonso J. Agudo Pérez. Avda. Doctor Federico Rubio, 106. Madrid 28040.


Y a la venta en: EL CORTE INGLES – GALERIAS PRECIADOS – PRYCA – CONTINENTE – HYPERCORD y establecimientos especializados.
MAX HEADROOM
Lo que yo quisiera saber es . . .
Disponible en - CBM 64 SPECTRUM - 48/128 K y AMSTRAD.
Porqué alguien querrá jugar con algo diferente?
QUICKSILVA
Los juegos más poderosos del universo
MIND GAMES ESPAÑA S.A
MARIANO CUBI, 4 ENTLO.
TEL. 218 34 00
08006 BARCELONA
© Chrysalis Visual Programming Ltd.